PHD

Investigations on flight trajectory optimisation and adaptive control

MacCormac, J. K. M.

Award date:
1994

Awarding institution:
University of Bath

Link to publication

Investigations On Flight Trajectory Optimisation And Adaptive Control.

Submitted by

J.K.M. MacCormac B.Sc.,C.Eng.,F.R.Ae.S.,F.Zw.I.E.,M.I.E.E.

for the degree of Ph.D.

of

The University of Bath

1994.

Dedication

This thesis is presented in recognition of the support and encouragement provided by many individuals throughout my career.

It is offered as a small contribution in the pursuit of excellence in flight control system performance, and in the hope that closed-loop adaptive flight control systems will one day be the norm.

The thesis is dedicated to that incredible breed of men, namely the Test Pilots, without whom no progress in flight system research is possible.

In particular I give recognition to those who have made significant contributions to the success of the following projects, and with whom I have been privileged to work.

V.C. 10 and B.A.C.111 Auto-land Programme.	E. McNamara, B.Trubshaw, G. Corps.
W.G.13 - Lynx Helicopter Flight Test Programme.	W.R. Gellatley, L.R. Moxam, J.G.P. Morton.
Multi-Role Combat Aircraft - Tornado Flight Test Programme.	P. Millett, Hr. Meister, Sgn. Trevisan.

Summary

The application of Optimal Control theory to define optimal trajectory manoeuvres for high performance aircraft results in the definition of the optimal control variables and nominal state variable responses that optimise some performance index.

In this thesis the specific problem of an aircraft acquiring maximum height in a fixed time while satisfying a specified terminal constraint - namely that of the finally achieved velocity - and also minimising a function of drag, is investigated.

The generalised necessary conditions for optimal control are defined in chapter 1. Chapter 2 applies these necessary conditions to the specific optimisation problem cited above.

Chapter 3 is concerned with the numerical solution of the equations resulting from the necessary conditions and in particular the solution of the resultant two point boundary value problems. Two numerical methods for the solution of two point boundary value problems are presented. These are respectively the method of Steepest Descent and in chapter 4 the method of Quasilinearisation.

For this optimisation process the control variable has been chosen as the angle of attack or aerodynamic incidence 'α', and the solution of the optimisation process results in the definition of the optimal control α^* together with the optimal state variables namely velocity V^*, flight path angle γ^*, and mass m^*.

For the aircraft to follow this optimum trajectory it is necessary to control the aircraft about its Pitch axis by the application of elevator such that the optimum value of angle of attack is achieved. The required value of pitch rate is readily derived from the relationship $q^* = \dot{\gamma}^* - \dot{\alpha}^*$.

To achieve this optimum value of pitch rate a command stability augmentation system is required as the elevator response characteristics of the aircraft vary significantly throughout the optimum trajectory.

Just how the aircraft pitch rate responses to elevator vary on the optimal trajectory is defined in chapter 5.

As the dynamics of the aircraft are changing it is necessary to design the command-stability augmentation system to compensate for these changes in dynamics, in order that the response of pitch rate achieved to pitch rate demand is maintained acceptable throughout the optimal manoeuvre. This process of modifying the elevator controller is known as adaptation of the controller and in chapter 6 a review of adaptive techniques is provided.

The variation of aircraft parameters is such that a unique relationship between the parameters and auxiliary variables such as speed, mach number, dynamic pressure or altitude does not readily exist and this would result in a complex open loop adaptive gain scheduling scheme. Chapter 7 details the design of a closed loop Indirect Adaptive Command Stability Augmentation system with particular emphasis on the identification of the time varying uncertain parameters of the aircraft. The identification procedure is treated as a two point boundary value problem and the method of Quasilinearisation is again applied to the solution of this.

Consideration is given to the robustness of the controller and in particular emphasis on establishing a degree of confidence in the identified parameters is presented in chapter 8. The thesis concludes with suggestions for further work and on practical considerations relating to the implementation of this on line adaptive controller.

Results of the generation of the optimal trajectory, the variation of the aircraft parameters, the on-line identification of the parameters, and the response of the adaptive controller are included at the end of the appropriate chapters. The programmes written to obtain these results are presented in the appendices.

Contents

Introduction

This thesis is concerned with the design procedure for an on-line closed loop adaptive command stability augmentation system suitable for use on high performance aircraft. The system investigated is an alternative solution to the traditional open loop controller gain scheduling adaptive systems currently utilised to cater for the variation in aircraft dynamics throughout the flight envelope.

It has been stated that auxiliary gain scheduling is adequate for most flight situations and indeed the author in no way wishes to cast aspersions on this technique which has served the design of flight control systems well. There are however some modes of operation of aircraft where it is difficult to obtain a unique relationship between controller parameters and flight condition states. A case in point is the optimal climb manoeuvre where speed, height, dynamic pressure are all varying widely throughout the optimal trajectory.

A specific optimisation problem, namely that of acquiring maximum height in a fixed time, while minimising aerodynamic drag and achieving a specified terminal constraint on the velocity at the end of the optimal manoeuvre, has been chosen as the starting point for this investigation on adaptive control.

The motivation for this starting point is three fold. First, the optimal solution to this problem requires the aircraft to traverse a significant portion of the flight envelope and in so doing encounters significant aerodynamic parameter changes. Secondly, the solution of this problem entails the application of optimal control theory and it is a long term objective of the author to incorporate a degree of optimisation into the adaptive process in order to create an optimal adaptive controller. Finally, one of the techniques used for the numerical solution of the two point boundary value problem which arises in the optimisation process is to be employed in the solution of the on-line identification of the aircraft dynamics as required for the indirect adaptive system . The identification of the changing system parameters is treated as an additional two point boundary value problem which is solved by Quasilinearisation techniques in a manner similar to that applied in the optimisation problem. By this step by step approach the author has gained experience and confidence in the application of these techniques.

A fixed time optimisation problem, as opposed to one where time is to be explicitly optimised, as for example in the minimum time to climb problem, has been considered purely for computational simplicity whereby the added complexity of modifying the optimal time, by the solution of the Transversality condition, is avoided.

Early attempts at closed loop adaptive flight control were based on direct adaptive control techniques where the dynamic parameters of the aircraft were not explicitly determined. These methods were rather unsuccessful and resulted in the catastrophic failure of a research aircraft on an adaptive control test flight.

Although much research has been undertaken in closed loop adaptive control techniques with regard to stability and robustness of the control in the past two decades, these methods have not been applied in production flight control systems design. Instead system designs have opted for an open loop form of adaptive control whereby controller parameters are scheduled as a function of some auxiliary variable; for example Mach number, dynamic pressure, height, or incidence.

The objective of adaptive control, be it open loop or closed loop, is to retain the handling qualities of the aircraft within acceptable limits throughout the flight envelope. It is a debatable point whether a uniform identical response characteristic is desirable at all flight conditions. To demonstrate the principle of closed loop adaptive control this unchanging response characteristic has been taken as an objective for the purposes of this investigation. Should a variation in handling qualities be required at different points on the flight envelope the system will also be capable of adapting to these changing performance criteria.

Gain scheduling of controller parameters by auxiliary variables has served the industry well and the importance of this technique should not be underestimated. However, as the goal of enhanced aircraft performance is relentlessly pursued, ever more complex algorithms for controller gain scheduling are required. This is in particular true if a unique relationship between the auxiliary scheduling variables and the required controller gains does not exist. While it is true that the implementation of complex scheduling algorithms is now more readily facilitated with the significant improvement in digital signal processing technology that has occurred over the past two decades, it can also be argued that the availability of these devices now facilitates the implementation of robust closed loop adaptive systems. This then is a further motivation for this investigation at this time.

A review of adaptive control techniques is presented in chapter 6 . This investigation has concentrated on the so called indirect adaptive control technique where the dynamics of the aircraft to be controlled are specifically identified. A recursive on line continuous identification process is incorporated in the adaptive system to reduce the uncertainty of the plant parameters. The author has a preference for this approach to adaptive control, if for no other reason than a conviction that the more that is known about the current parametric representation of the aircraft dynamics the more erudite the generation of the appropriate control action.

On-line identification requires the dynamics of the system to be identified to be stimulated before estimates of the unknown parameters can be determined. This is known as the Persistency of Excitation requirement. In most control applications it is undesirable to introduce additional extraneous control perturbation signals in order to excite the system purely for the purposes of identification. This is because in general these additional 'test' signals will contaminate the system response. A further objective then is to determine estimates of the system dynamics using the normal control inputs to the system. For this to be possible the control input must be sufficiently 'rich' in excitation frequencies to excite all the modes of the system. The normal operating control inputs may not satisfy these requirements at all times. In this event accurate system identification is not possible. Rather than augment the stimuli to the system to fulfil the identification requirements, the approach in this study has been to determine when the identification estimates are unreliable and to inhibit the adaptive update of the controller parameters should this occur.

This has proved to be satisfactory for the particular problem investigated. The procedure is based on the alternative philosophy that if specific modes are not excited there is no need to adapt the controller to cater for them.

Considerable attention has therefore been focused on determining when the confidence level of the accuracy of the system identification is low, in order to inhibit the adaptation process during this period. The adaptation of the controller resumes automatically as soon as the identification process is again satisfactory.

In this approach the view has been taken that as long as the system response is satisfying the demand then the overall performance is satisfactory. During the period that the adaptation is inhibited it is possible for a mismatch to occur between the controller parameters and the dynamics of the system being controlled. In this event the combination of controller and system parameters may well deviate from the chosen optimum criterion. The system will not initially respond to a subsequent command in precisely the manner required; however this command will reactivate the identification process and appropriate adaptation of the controller. As the identification and adaptation take a finite time the system response may well have deviated from the nominal desired response even though the controller parameter and system parameter combination will be correctly matched at the end of the first identification and adaptation interval. This means that the combined adaptive controller and system combination will have the correct dynamic response characteristics in terms of overall closed loop system parameters but the actual output response trajectory may deviate from the nominal due to the initial mismatch of controller parameters. This situation can be alleviated by augmenting the control ,now that the system dynamics have been identified, such that the original response trajectory is regained in an optimum manner. If this further requirement is incorporated in the control scheme the controller is in effect an optimum adaptive controller.

Chapter 1

OPTIMAL CONTROL

Historical Background.

Optimal Control is basically the selection of a set of control variables from the set of admissible controls which either maximise a performance index or minimise a cost functional. An admissible control is one which lies within and does not invade the control constraints during the period of optimum performance - the optimisation interval $t_f - t_0$

A mathematical technique available to perform the optimal control selection is based on the Calculus of Variations. Historically these techniques have been available since the time of the ancient Greeks, however it was not until the time of Sir Isaac Newton and Johann Bernoulli in the late seventeenth century that the mathematics were formally developed. Both of these mathematicians considered and solved the brachistochrone problem which requires the shape of a frictionless wire with fixed end points to be defined such that a bead may slide from one end of the wire to the other, under gravity, in minimum time. The solution - a cycloid - of this seemingly obscure problem has much relevance today in the re-entry of space craft to the earth's atmosphere, a fact which the author finds both incredible and fascinating. There are many interesting texts on the subject of the calculus of variations. References (1,2,&3) the author has found to be invaluable to gain an understanding of the subject.

Bryson (2) has formally derived the necessary conditions for optimal control in a succinct fashion and the author can do no better than refer the reader to this excellent work. The derivations are for this reason omitted from the thesis, however the necessary conditions for optimal control are presented as follows and applied in a subsequent chapter.

The Necessary Conditions For Optimal Control

The Performance Index or Cost Function:-

$$J = \phi(\underline{x}(t_f),t_f) + \underline{\gamma}^T \psi(\underline{x}(t_f),t_f) + \int_{t_0}^{t_f} \{L(\underline{x},\underline{u},t) + \underline{\lambda}^T (f(\underline{x},\underline{u},t) - \dot{\underline{x}})\} dt. \quad (1)$$

Where:-

$\dot{\underline{x}}(t) = f(\underline{x},\underline{u},t)$ represents the state equations of the n-dimensional dynamical system to be optimally controlled.

$\underline{\lambda}(t)$ represents the n-dimensional co-state vector. This vector introduces a cost factor on the system dynamics not being satisfied.

$L(\underline{x},\underline{u},t)$ is the the integrand function to be optimised over the interval t_0 to t_f.

$\phi(\underline{x}(t_f),t_f)$ is the function of terminal conditions to be optimised.

$\psi(x(t_f),t_f) = 0$ is the vector function of specified terminal state constraints to be satisfied in the optimisation problem.

γ^T is the weighting vector on the terminal constraints.

The minimisation of a cost functional of the form of (1) results in the requirement that the following necessary conditions are satisfied.

The Euler-Lagrange Equations:-

$$\dot{\underline{\lambda}} = -(\frac{\partial f}{\partial x})^T \underline{\lambda} - (\frac{\partial L}{\partial x})^T$$

$$(\frac{\partial f}{\partial x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \vdots & \vdots & \frac{\partial f_1}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial x_1} & \vdots & \vdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \text{ and } (\frac{\partial L}{\partial x}) = \begin{bmatrix} \frac{\partial L}{\partial x_1} & \vdots & \vdots & \frac{\partial L}{\partial x_n} \end{bmatrix}$$

The Optimality Condition:-

$$(\frac{\partial L}{\partial u}) + \underline{\lambda}^T (\frac{\partial f}{\partial u}) = 0$$

$$(\frac{\partial L}{\partial u}) = \left[\frac{\partial L}{\partial u_1} \;\vdots\; \cdots \;\vdots\; \frac{\partial L}{\partial u_m} \right] \text{ and } (\frac{\partial f}{\partial u}) = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}$$

The Optimal Control is obtained from the solution of the optimality equation. At each time step in the integration process the vectors $\underline{x}(t)$ and $\underline{\lambda}(t)$ are substituted in the optimality condition to give the optimal control vector $\underline{u}^*(t)$.

Constrained Controls:

If the controls are constrained such that

$$u_{\min} \leq u^* \leq u_{\max}$$

then the Optimality Condition becomes

$$\{(\frac{\partial L}{\partial u}) + \underline{\lambda}^T (\frac{\partial f}{\partial u})\}\delta u \geq 0$$

This is Pontryagin's Minimum Principle.

The Boundary Conditions:-

(a) Co-State Terminal conditions.

$$\underline{\lambda}^T(t_f) = (\frac{\partial \phi}{\partial x}) + \gamma^T (\frac{\partial \psi}{\partial x})$$

$$(\frac{\partial \phi}{\partial x}) = \left[\frac{\partial \phi}{\partial x_1} \vdots \quad \vdots \quad \vdots \quad \vdots \frac{\partial \phi}{\partial x_n} \right]$$

and

$$(\frac{\partial \psi}{\partial x}) = \begin{bmatrix} \frac{\partial \psi_1}{\partial x_1} & \vdots & \vdots & \vdots & \frac{\partial \psi_1}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial \psi_q}{\partial x_1} & \vdots & \vdots & \vdots & \frac{\partial \psi_q}{\partial x_n} \end{bmatrix}$$

(b)State Initial conditions.

$$\underline{x}(t_0) = specified$$

The Transversality Condition:-

$$\{\frac{\partial \phi}{\partial t} + \underline{\gamma}^T (\frac{\partial \psi}{\partial t}) + \underline{\lambda}^T (\dot{\underline{x}}) + \mathsf{L}\}\Big|_{t_f} = 0$$

The transversality condition determines the final time of the optimisation interval in the case where the final time is not fixed.

State Inequality Constraints:-

State inequality constraints of the form

$$G(x,u,t) \leq 0$$

can be included in the optimisation problem by augmenting the order n of the number of state differential equations, and forming an additional equation

$$\dot{x}_{n+1} = K\{G(\underline{x},\underline{u},t)\}^2$$

K is assigned the value zero if the constraint is not violated or a large positive constant if the optimum state trajectory invades the state constraint boundary.

Non-Linear Boundary Value Problem:-

The simultaneous solution of the state and co-state non-linear differential equations in general constitutes a non-linear two-point boundary value problem. The initial conditions of the states are usually known, however only the terminal conditions of the co-states are known.

Methods of solution:-

Two methods of solution of the resultant two-point boundary value problem are investigated. These are the method of 'Steepest Descent' and the method of 'Quasilinearisation" presented in chapters three and four respectively.

Chapter 2

The Maximum Climb In A Fixed Time:- Optimisation Problem.

For the purpose of this thesis, the optimisation problem of acquiring maximum height in a fixed time, while attaining a terminal constraint on the final velocity, and minimising a function of drag, has been investigated. This problem has been selected as in its solution the aircraft encounters a large number of different flight conditions in the flight envelope. Hence when the optimal trajectory is flown, a significant variation in aircraft system parameters will be encountered. This will enable the effectiveness of the on-line adaptive command stability augmentation system to be assessed.

The non-linear equations of motion governing the system dynamics are obtained with reference to fig.1. There are five state equations representing the system dynamics and these are presented below.

The System Dynamics.

STATE VARIABLES:-

$$\underline{\dot{x}} = f(\underline{x}, \underline{u}, t)$$

Symbols:-

Longitudinal Acceleration	$\dot{V}$	$ft.\mathrm{sec}^{-2}$
Flight path Angular Velocity	$\dot{\gamma}$	$rad.\mathrm{sec}^{-1}$
Height Rate	$\dot{h}$	$ft.\mathrm{sec}^{-1}$
Horizontal Velocity	$\dot{x}$	$ft.\mathrm{sec}^{-1}$
Rate of change of mass	$\dot{m}$	$slug\ \mathrm{sec}^{-1}$
Control Variable	$u \equiv \alpha$	$rad.$

The State Equations:-

$$\dot{V} = \frac{1}{m}\{T\cos\alpha - D - mg\sin\gamma\} \qquad \ldots\ldots f_1$$

$$\dot{\gamma} = \frac{1}{mV}\{L + T\sin\alpha - mg\cos\gamma\} \qquad \ldots\ldots f_2$$

$$\dot{h} = V\sin\gamma \qquad \ldots\ldots f_3$$

$$\dot{x} = V\cos\gamma \qquad \ldots\ldots f_4$$

$$\dot{m} = -\frac{T}{cg} \qquad \ldots\ldots f_5$$

L

Lift

$$L(V,h,\alpha) = \frac{1}{2}\rho V^2 S C_L$$

Drag

$$D(V,h,\alpha) = \frac{1}{2}\rho V^2 S C_D$$

Coefficient of Lift

$$C_L(M,\alpha) = C_{L_\alpha}(M)\alpha$$

Coefficient of Drag

$$C_D(M,\alpha) = C_{D_0} + \eta C_{L_\alpha}^2 \alpha^2$$

The appropriate values of $C_{D_0}(M), C_{L_\alpha}(M), \eta(M)$ are obtained from fig.2 by interpolation at the corresponding Mach number.

Air Density

$$\rho(h) = \rho_0 e^{-\frac{h}{h_1}} \quad \rho_0 = 0.00254 slug\,ft^{-3} \quad h_1 = 27300 ft.$$

Mach Number $$M(a,V) = \frac{V}{a}$$

Local Speed of Sound $$a(t_h) = a_{S.L.}\sqrt{\frac{t_h}{t_0}}$$

Sea Level

$$a_{S.L.} = 331.46 meters\ \sec^{-1} \equiv 1087.47 ft.\ \sec^{-1}$$

I.C.A.O. Standard Atmosphere.

$$t_h = \begin{cases} 288.15^0K & S.L. \\ (288.15 - k_h h)^0K & 0 \le h \le 11Km. \equiv 0 \le h \le 36089 ft. \\ 216.65^0K & 11Km. \le h \le 20Km. \equiv 36089 \le h \le 65617 ft. \end{cases}$$

Lapse Rate $$k_h = 6.5^0C.Km^{-1} \equiv 0.00198^0C.ft.^{-1}$$

Fig. 3 shows how the speed of sound varies as a function of height on the I.C.A.O. Standard Atmosphere h V a graph. At each height on the optimal trajectory computations, the local value of air density is computed together with the value of the local speed of sound. From a knowledge of the longitudinal velocity, the corresponding mach number can then be evaluated. The current value of mach number is then used to interpolate the aerodynamic data and thrust.

Thrust $T(h,M)$ is interpolated from the engine characteristics as defined in fig. 4 as a function of both height and mach number.

Gravity $$g = 32.2 ft.\sec^{-2}$$

Surface Reference Area $$S = 530 ft^2$$

Engine Specific Impulse $c = 1600\,\text{sec.}$

Investigation of Maximum Climb in a Fixed Time with Minimum Drag.

Specified Terminal Constraints:-

$$V(t_f) = 968.58\,ft.\sec^{-1} \equiv M = 1.0 \; for \; h \geq 36000\,ft.$$
$$\therefore \; \psi_1 = V(t_f) - 968.58$$

Integral Function to be Minimised:-

$$\frac{1}{2}\int_{t_0}^{t_f} u^2 dt. = \frac{1}{2}\int_{t_0}^{t_f} \alpha^2 dt.$$

Specified Initial Conditions:-

$$\underline{x}(t_0) = \begin{cases} V(t_0) = 400.0\,ft.\sec^{-1} \\ \gamma(t_0) = 0.0\,rad. \\ h(t_0) = 700.0\,ft. \\ x(t_0) = 0.0\,ft. \\ m(t_0) = 42000.\,lb. \equiv 1304.35\,Slugs. \end{cases}$$

Terminal Quantity To Be Optimised:-

$$\phi(\underline{x}(t_f), t_f) = -h(t_f)$$

The negative sign implies that the minimisation of this terminal quantity will result in the maximisation of $h(t_f)$.

The Specific Cost Function:-

The resultant cost function to be minimised for this maximum height in a fixed time problem is given by:-

$$J = -h(t_f) + \gamma_1 (V(t_f) - 968.58) + \frac{1}{2}\int_{t_0}^{t_f} \alpha^2 dt.$$

The Elements $(\frac{\partial f}{\partial x})$ **:-**

$$\frac{\partial f_1}{\partial V} = \frac{1}{m}(\frac{\partial T}{\partial V}\cos\alpha - \frac{\partial D}{\partial V}) \;\;.,\;\; \frac{\partial f_1}{\partial \gamma} = -g\cos\gamma \;\;.,$$

$$\frac{\partial f_1}{\partial h} = \frac{1}{m}(\frac{\partial T}{\partial h}\cos\alpha - \frac{\partial D}{\partial h})$$

$$\frac{\partial f_1}{\partial x} = 0 \;\;.,\;\; \frac{\partial f_1}{\partial m} = \frac{D - T\cos\alpha}{m^2}$$

$$\frac{\partial f_2}{\partial V} = \frac{1}{mV^2}\{V(\frac{\partial T}{\partial V}\sin\alpha + \frac{\partial L}{\partial V}) - (L + T\sin\alpha - mg\cos\gamma)\}$$

$$\frac{\partial f_2}{\partial \gamma} = \frac{g}{V}\sin\gamma \;\;.,\;\; \frac{\partial f_2}{\partial h} = \frac{1}{mV}(\frac{\partial T}{\partial h}\sin\alpha + \frac{\partial L}{\partial h}) \;\;.,\;\; \frac{\partial f_2}{\partial x} = 0$$

$$\frac{\partial f_2}{\partial m} = -\frac{1}{m^2 V}(L + T\sin\alpha)$$

$$\frac{\partial f_3}{\partial V}=\sin\gamma \;.,\; \frac{\partial f_3}{\partial \gamma}=V\cos\gamma \;.,\; \frac{\partial f_3}{\partial h}=0 \;.,\; \frac{\partial f_3}{\partial x}=0 \;.,\; \frac{\partial f_3}{\partial m}=0$$

$$\frac{\partial f_4}{\partial V}=\cos\gamma \;.,\; \frac{\partial f_4}{\partial \gamma}=-V\sin\gamma .,\; \frac{\partial f_4}{\partial h}=0 .,\; \frac{\partial f_4}{\partial x}=0 \;.,\; \frac{\partial f_4}{\partial m}=0$$

$$\frac{\partial f_5}{\partial V}=-\frac{1}{cg}\frac{\partial T}{\partial V} \;.,\; \frac{\partial f_5}{\partial \gamma}=0.,\; \frac{\partial f_5}{\partial h}=-\frac{1}{cg}\frac{\partial T}{\partial h} \;.,\; \frac{\partial f_5}{\partial x}=0 .,\; \frac{\partial f_5}{\partial m}=0$$

The Elements of $(\frac{\partial L}{\partial x})$:-

The Lagrangian $L=\frac{1}{2}\alpha^2$ is independent of the system state variables.
Hence

$$(\frac{\partial L}{\partial x})=[0 \;\vdots\; 0 \;\vdots\; 0 \;\vdots\; \;\;\vdots\; 0]$$

Substitution of the above defined elements into the Euler Lagrange equations results in the following set of co-state equations.

The Co-State Equations

$$\begin{bmatrix} \dot{\lambda}_V \\ \dot{\lambda}_\gamma \\ \dot{\lambda}_h \\ \dot{\lambda}_x \\ \dot{\lambda}_m \end{bmatrix} = - \begin{bmatrix} \frac{1}{m}(\frac{\partial T}{\partial V}\cos\alpha - \frac{\partial D}{\partial V}) & \frac{1}{mV^2}\{V(\frac{\partial T}{\partial V}\sin\alpha + \frac{\partial L}{\partial V}) - (L + T\sin\alpha - mg\cos\gamma)\} & \sin\gamma & \cos\gamma & -\frac{1}{cg}\frac{\partial T}{\partial V} \\ -g\cos\gamma & \frac{g}{V}\sin\gamma & V\cos\gamma & -V\sin\gamma & 0 \\ \frac{1}{m}(\frac{\partial T}{\partial h}\cos\alpha - \frac{\partial D}{\partial h}) & \frac{1}{mV}(\frac{\partial T}{\partial h}\sin\alpha + \frac{\partial L}{\partial h}) & 0 & 0 & -\frac{1}{cg}\frac{\partial T}{\partial h} \\ 0 & 0 & 0 & 0 & 0 \\ \frac{D - T\cos\alpha}{m^2} & -\frac{1}{m^2 V}(L + T\sin\alpha) & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_V \\ \lambda_\gamma \\ \lambda_h \\ \lambda_x \\ \lambda_m \end{bmatrix}$$

The Optimal Control:-

From the optimality condition $(\frac{\partial \mathrm{L}}{\partial u})+\underline{\lambda}^T(\frac{\partial f}{\partial u})=0$

$$\alpha+\left[\lambda_V \;\vdots\; \lambda_\gamma \;\vdots\; \lambda_h \;\vdots\; \lambda_x \;\vdots\; \lambda_m\right]\begin{bmatrix} -\frac{1}{m}\{T\sin\alpha+\frac{\partial D}{\partial \alpha}\} \\ \frac{1}{mV}\{\frac{\partial L}{\partial \alpha}+T\cos\alpha\} \\ 0 \\ 0 \\ 0 \end{bmatrix}=0$$

This gives

$$\alpha-\frac{\lambda_V}{m}\{T\sin\alpha+\frac{\partial D}{\partial \alpha}\}+\frac{\lambda_\gamma}{mV}\{\frac{\partial L}{\partial \alpha}+T\cos\alpha\}=0 \qquad \ldots\ldots g_1$$

from which the optimal control is determined.

Note:- $g_1(V,h,m,\lambda_V,\lambda_\gamma,\alpha)=0$

The Terminal Conditions On The Costates:-

$$\lambda_V(t_f)=\gamma_1$$

$$\lambda_\gamma(t_f)=0$$

$$\lambda_h(t_f)=-1$$

$$\lambda_x(t_f)=0$$

$$\lambda_m(t_f)=0$$

The State and Costate Equations.

$$\dot{V} = \frac{1}{m}\{T\cos\alpha - D - mg\sin\gamma\} \qquad \ldots\ldots f_1$$

$$\dot{\gamma} = \frac{1}{mV}\{L + T\sin\alpha - mg\cos\gamma\} \qquad \ldots\ldots f_2$$

$$\dot{h} = V\sin\gamma \qquad \ldots\ldots f_3$$

$$\dot{x} = V\cos\gamma \qquad \ldots\ldots f_4$$

$$\dot{m} = -\frac{T}{cg} \qquad \ldots\ldots f_5$$

$$\dot{\lambda}_V = \frac{\lambda_V}{m}(\frac{\partial D}{\partial V} - \frac{\partial T}{\partial V}\cos\alpha) + \frac{\lambda_\gamma}{mV^2}\{(L + T\sin\alpha - mg\cos\gamma) - V(\frac{\partial T}{\partial V}\sin\alpha + \frac{\partial L}{\partial V})\} - \lambda_h\sin\gamma - \lambda_x\cos\gamma + \frac{\lambda_m}{cg}\frac{\partial T}{\partial V} \qquad \ldots\ldots f_6$$

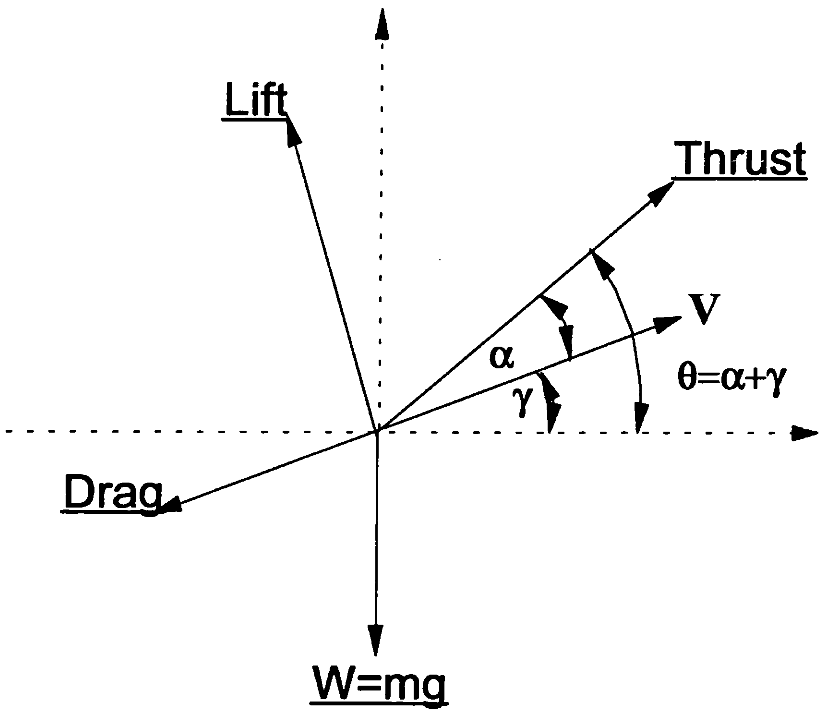
$$\dot{\lambda}_\gamma = \lambda_V g\cos\gamma - \lambda_\gamma\frac{g}{V}\sin\gamma - \lambda_h V\cos\gamma + \lambda_x V\sin\gamma \qquad \ldots\ldots f_7$$

$$\dot{\lambda}_h = \frac{\lambda_V}{m}(\frac{\partial D}{\partial h} - \frac{\partial T}{\partial h}\cos\alpha) - \frac{\lambda_\gamma}{mV}(\frac{\partial T}{\partial h}\sin\alpha + \frac{\partial L}{\partial h}) + \frac{\lambda_m}{cg}\frac{\partial T}{\partial h} \qquad \ldots\ldots f_8$$

$$\dot{\lambda}_x = 0 \qquad \ldots\ldots f_9$$

$$\dot{\lambda}_m = \frac{\lambda_V}{m^2}(T\cos\alpha - D) + \frac{\lambda_\gamma}{m^2 V}(L + T\sin\alpha) \qquad \ldots\ldots f_{10}$$

Aircraft Forces Diagram



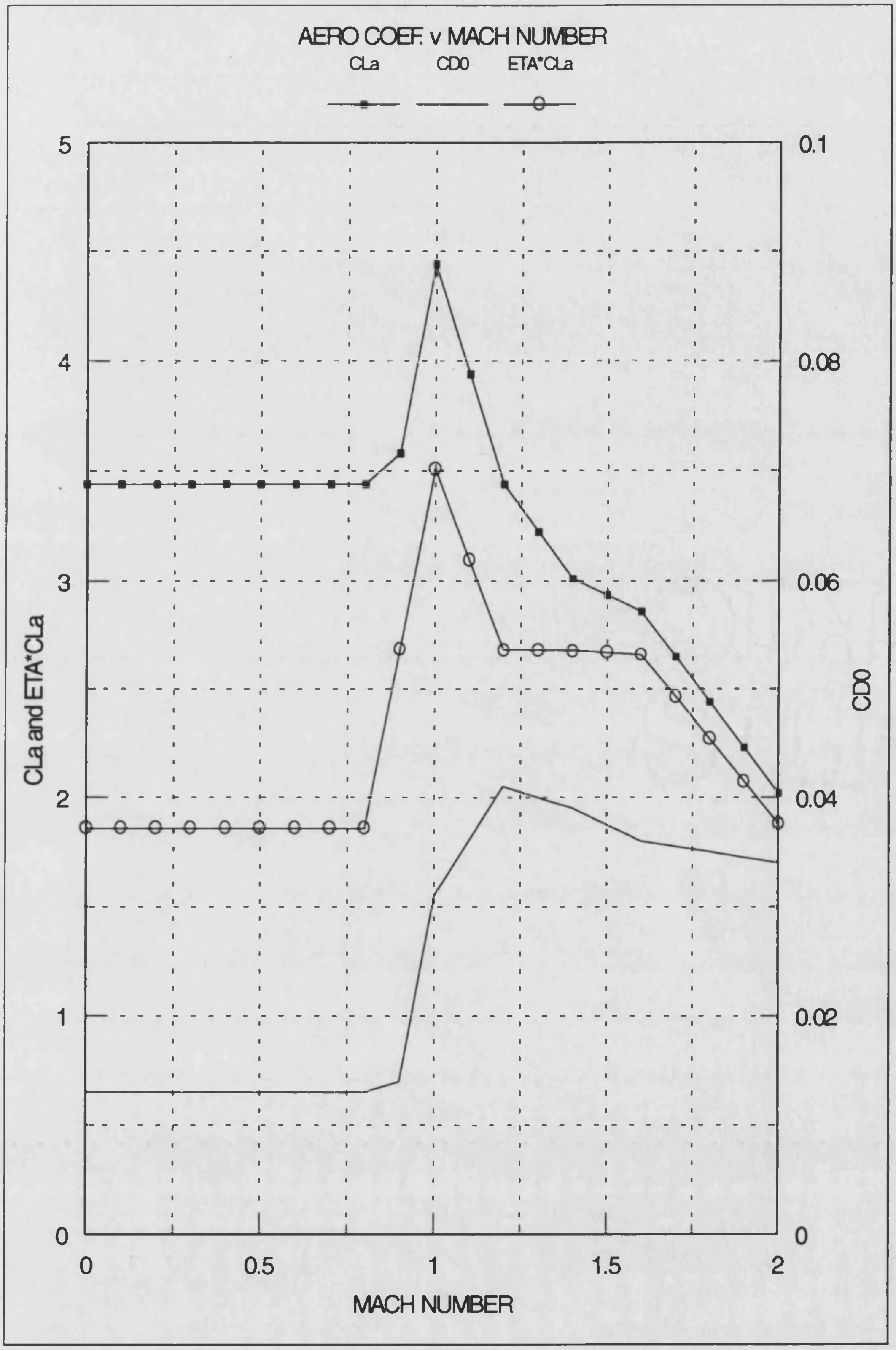
V = Velocity Along Flight Path

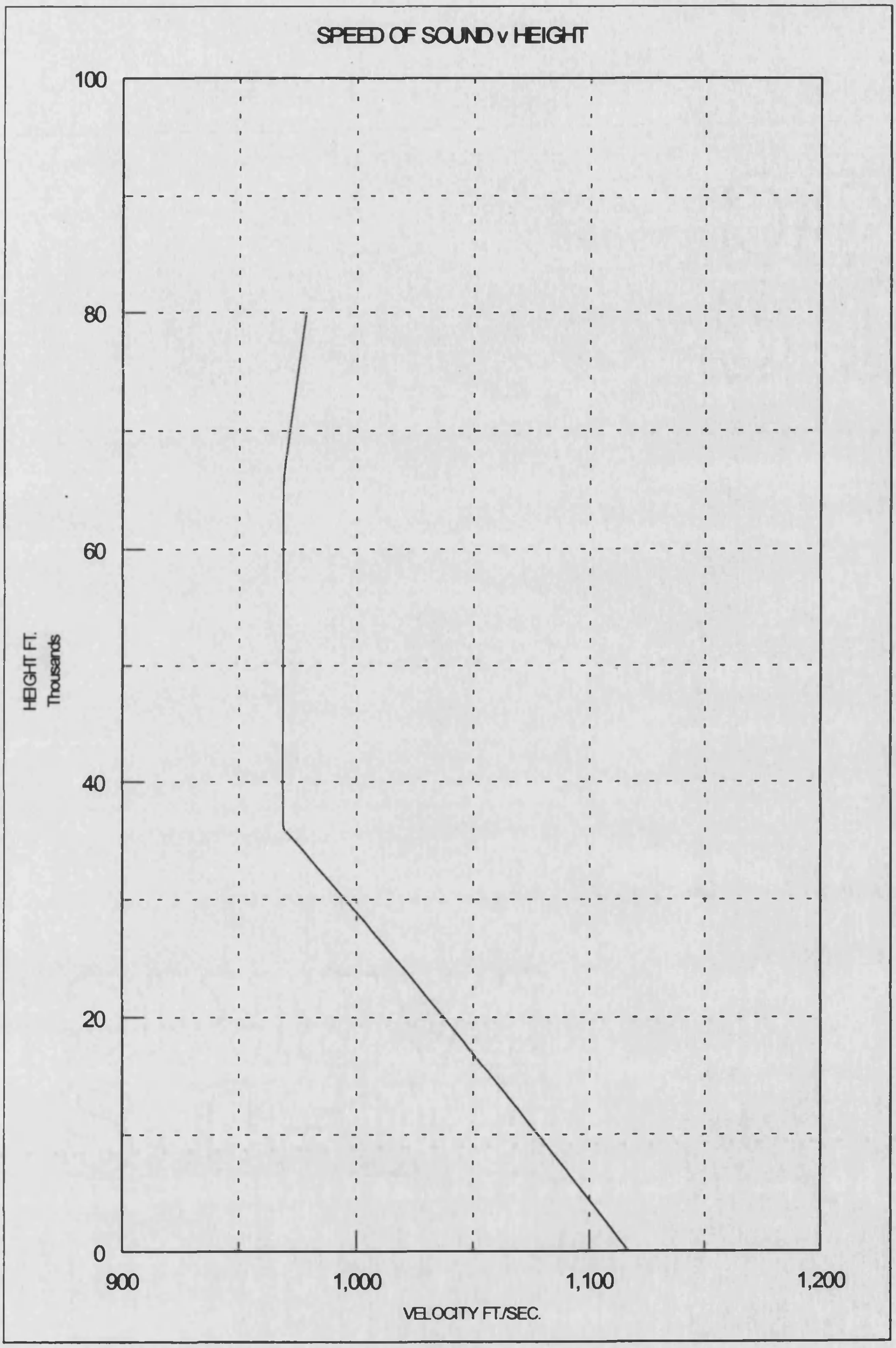
θ = Pitch Attitude Angle

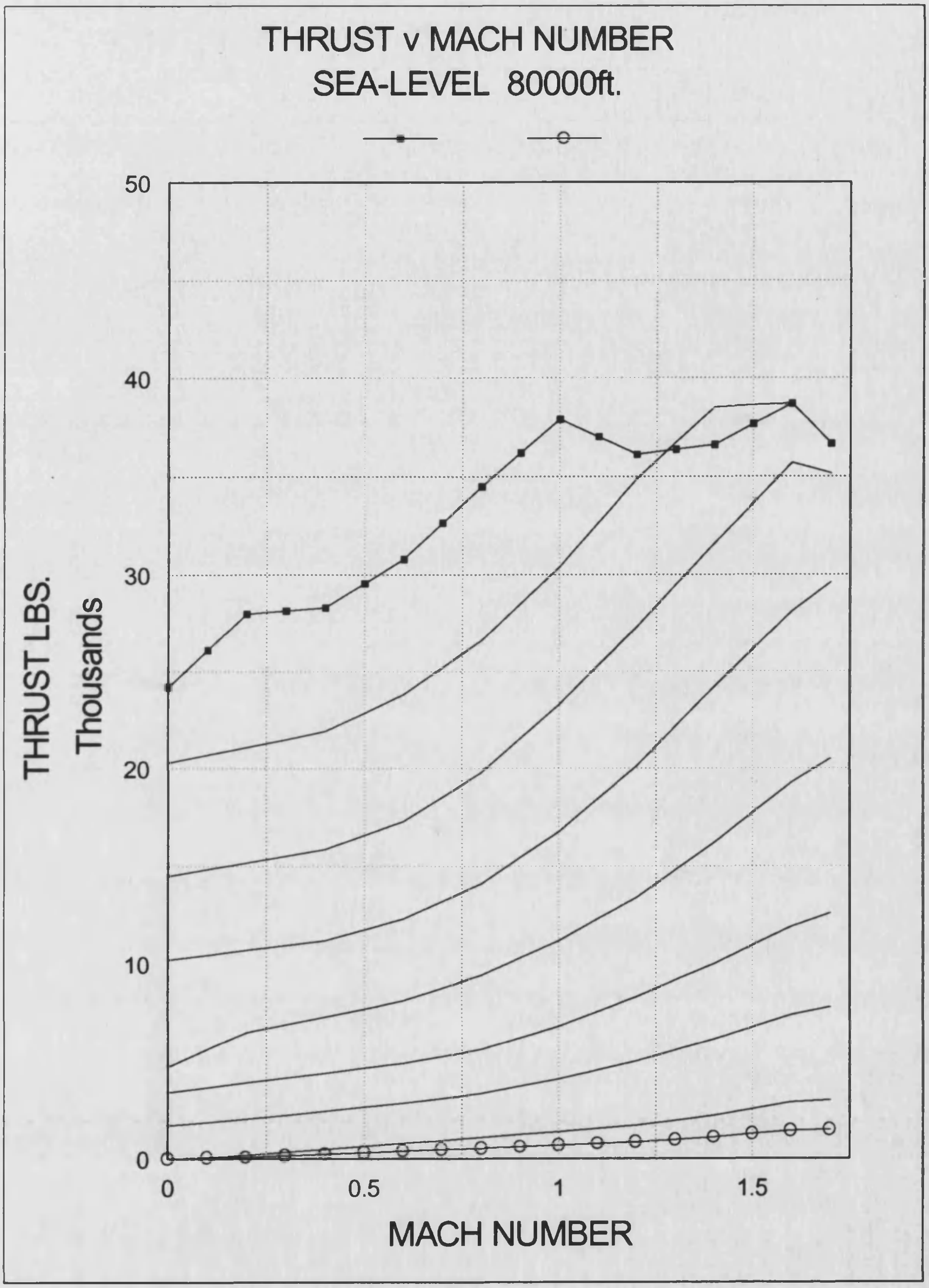
α = Angle of Attack

γ = Flight Path Angle

Fig. 1


AERO COEF. v MACH NUMBER
CLa
CD0
ETA*CLa
5
4
3
2
1
0
0.1
0.08
0.06
0.04
0.02
0
0
0.5
1
1.5
2
CLa and ETA*CLa
CD0
MACH NUMBER
Fig. 2


SPEED OF SOUND v HEIGHT
100
80
60
40
20
0
HEIGHT FT.
Thousands
900
1,000
1,100
1,200
VELOCITY FT/SEC.
Fig 3



THRUST v MACH NUMBER
SEA-LEVEL 80000ft.
THRUST LBS.
Thousands
50
40
30
20
10
0
0
0.5
1
1.5
MACH NUMBER

Fig.4

Chapter 3

Steepest Descent Solution Of The Optimisation Problem

The solution of the system state and co-state equations with appropriate boundary conditions, for the optimisation problem defined in chapter 2, is presented in this chapter. The solution is to be obtained by numerical computational techniques and will provide the optimal control - in this instance the optimum angle of incidence - time vector which will maximise the height acquired in a fixed time while satisfying a desired terminal constraint on the terminal velocity while minimising aerodynamic drag.

Two methods of solution of the optimisation equations are investigated. These are

a) The method of Steepest Descent described in this chapter ,and

b) The method of Quasilinearisation described in chapter 4.

Each of these computational procedures has its own relative merits in obtaining the optimised solution. The Steepest Descent method of solution has an initial advantage over the Quasilinearisation method in that so called starting vector solutions of both the state and co-state equations do not have to be estimated in order to initiate the iterative computational solution process. In the Steepest Descent method it is only necessary to estimate a starting vector for the optimal control time vector to commence the iterative procedure. The convergence of the algorithm is not too sensitive to the accuracy of this initial estimate and the method has good computational stability properties even when the control time vector estimate is significantly in error to the desired optimal control solution. It is also simpler to apply some engineering intuition in the selection of the initial control time vector estimate, than to provide estimates for all the state and co-state time vectors as required in the Quasilinearisation method of solution. The Steepest Descent method of solution however suffers from a linear convergence rate dependent on the gradient of the optimisation surface. As the gradient in general reduces as the optimum is approached this results in a comparatively slow rate of convergence to the true solution. This necessitates a large number of computational iterations before the optimum solution is obtained.

The Quasilinearisation algorithm on the other hand exhibits a quadratic convergence, but does require the initial set of starting vectors for the state and co-state variables to initiate the process. For this reason both methods of solution have been utilised in this investigation. The solutions obtained by the steepest descent method have been used as starting vectors for the Quasilinearisation method.

Steepest Descent Computational Procedure:-

The Steepest Descent computational procedure for the solution of the defined optimisation problem commences with the selection of an estimate for the control time vector $\alpha(t)$ over the optimisation interval t_0 to t_f. This initial control vector is shown in fig. 6 together with the converged solution of the optimal control obtained by the steepest descent method of solution.

The next step in the procedure is to perform a forward time integration of the five state equations starting from the initial conditions for the states as defined in chapter 2. A fourth order Runge-Kutta integration procedure was written for this purpose. At the end of the optimisation interval the complete set of state and co-state equations were integrated backwards in time from t_f to t_0. The terminal conditions used to initiate the backwards time integration process were as defined in chapter 2 for the co-state equations, while the values obtained from the forward time integration at the end of the optimisation interval were used for the state equations. In the case of $\lambda_V(t_f)$, this was set equal to a weighted function of the difference between the value obtained for $V(t_f)$ and the desired terminal constraint value of $968.58 ft.\sec^{-1}$ on the terminal velocity. This value corresponds to a terminal value of Mach 1 at the final height.

This choice of the terminal value of $\lambda_V(t_f)$ is obtained by a penalty function technique which seeks to minimise a weighted quadratic function of the error between the terminal velocity achieved and the desired terminal velocity constraint.

This weighted function can be expressed as

$$\frac{1}{2}\left[V(t_f)-V_D(t_f)\right]^T S\left[V(t_f)-V_D(t_f)\right]$$

This gives $\lambda_V(t_f)=\gamma_1 = S\left[V(t_f)-V_D(t_f)\right]$

At the end of the backward time integration a new updated control time vector is computed using the steepest descent algorithm. This is given by

$$u_{New} = u_{Old} - \tau\left[\frac{\partial L}{\partial u} + \underline{\lambda}^T(\frac{\partial f}{\partial u})\right]$$

where τ controls the displacement along the optimisation gradient from the current iteration control time vector to the control vector used in the next iteration. In the specific optimisation problem under consideration this becomes :-

$$\alpha_{N+1} = \alpha_N - \tau\{\alpha_N - \frac{\lambda_V}{m}(T\sin\alpha_N + \frac{\partial D}{\partial \alpha}) + \frac{\lambda_\gamma}{mV}(\frac{\partial L}{\partial \alpha} + T\cos\alpha_N)\}$$

where this expression is evaluated at each and every time step on the optimisation interval. If τ is set too large computing instabilities can ensue while if it is set too small the number of iterations required is significantly increased.

As a compromise τ was set to a small positive constant and successive iterations of the above procedure were performed until convergence was obtained. Convergence was determined by the change in control time vector being less than a small norm between successive iterations. After each convergence τ was simply doubled and the iterative process continued to a new convergence. Eventually it was observed that the improvement in maximum terminal height was reduced to a very low percentage of the absolute value on successive convergence. This was taken as the solution of the optimal control problem by the steepest descent method. It should be noted that at each integration time step in the optimisation interval the appropriate values of Lift, Drag and Thrust were computed together with the required partial derivatives which were automatically interpolated and evaluated from the aerodynamic and thrust data as a function of mach number and height.

The results of this Steepest Descent solution of the maximum climb optimisation problem are shown in fig. 5 as a function of Mach number. This should be compared with the initial height trajectory resulting from the control time vector estimate used to initiate the computational process.

The Steepest Descent optimisation programme written to obtain these results is presented in Appendix B.

The time vectors obtained of the state, co-state and optimal control were used as starting vectors for the Quasilinearisation computational procedure. This method of numerical solution of the optimisation problem is discussed in detail in the next chapter.

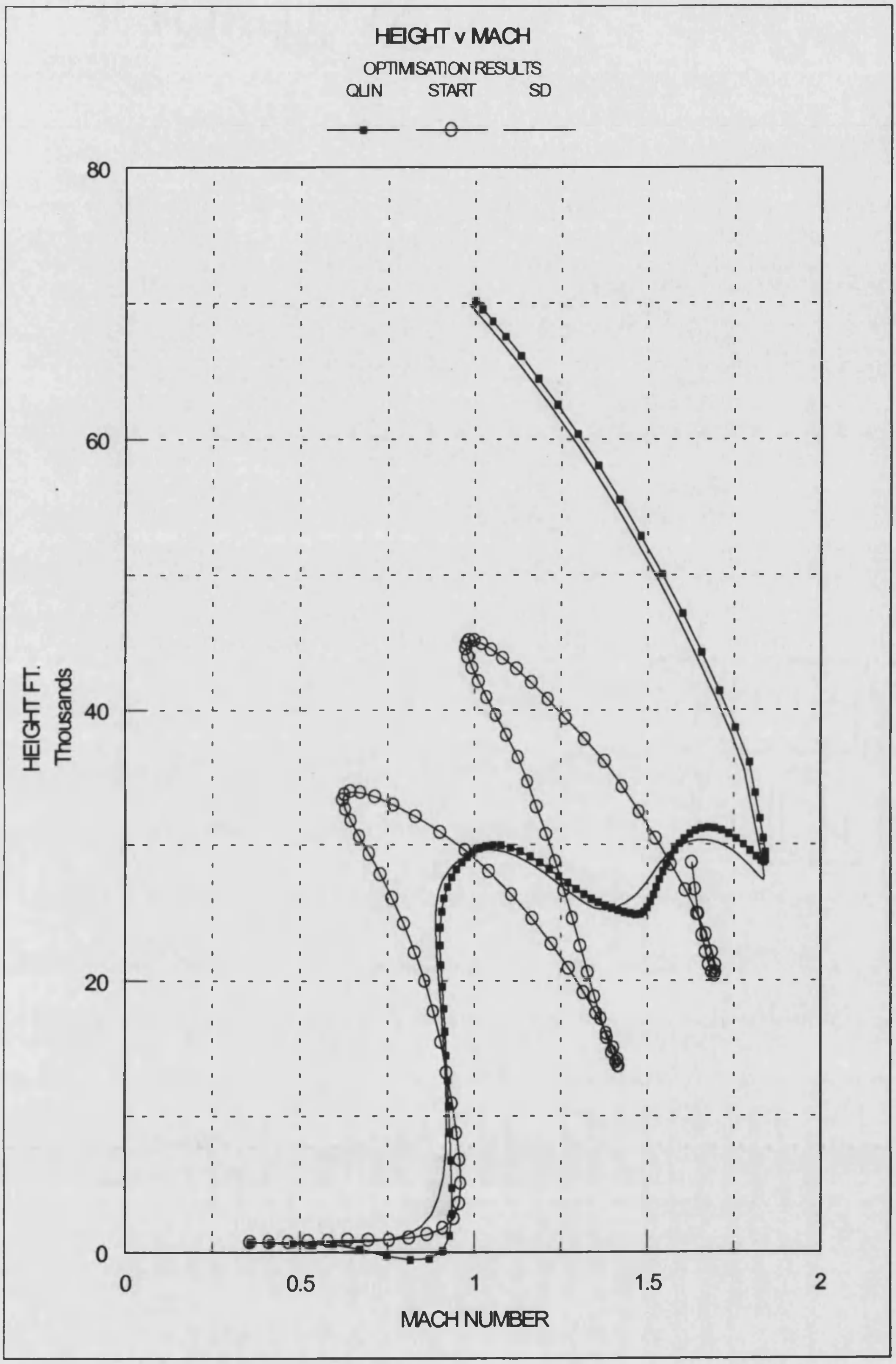

HEIGHT v MACH
OPTIMISATION RESULTS
QLIN
START
SD
HEIGHT FT.
Thousands
80
60
40
20
0
0
0.5
1
1.5
2
MACH NUMBER

Fig 5.

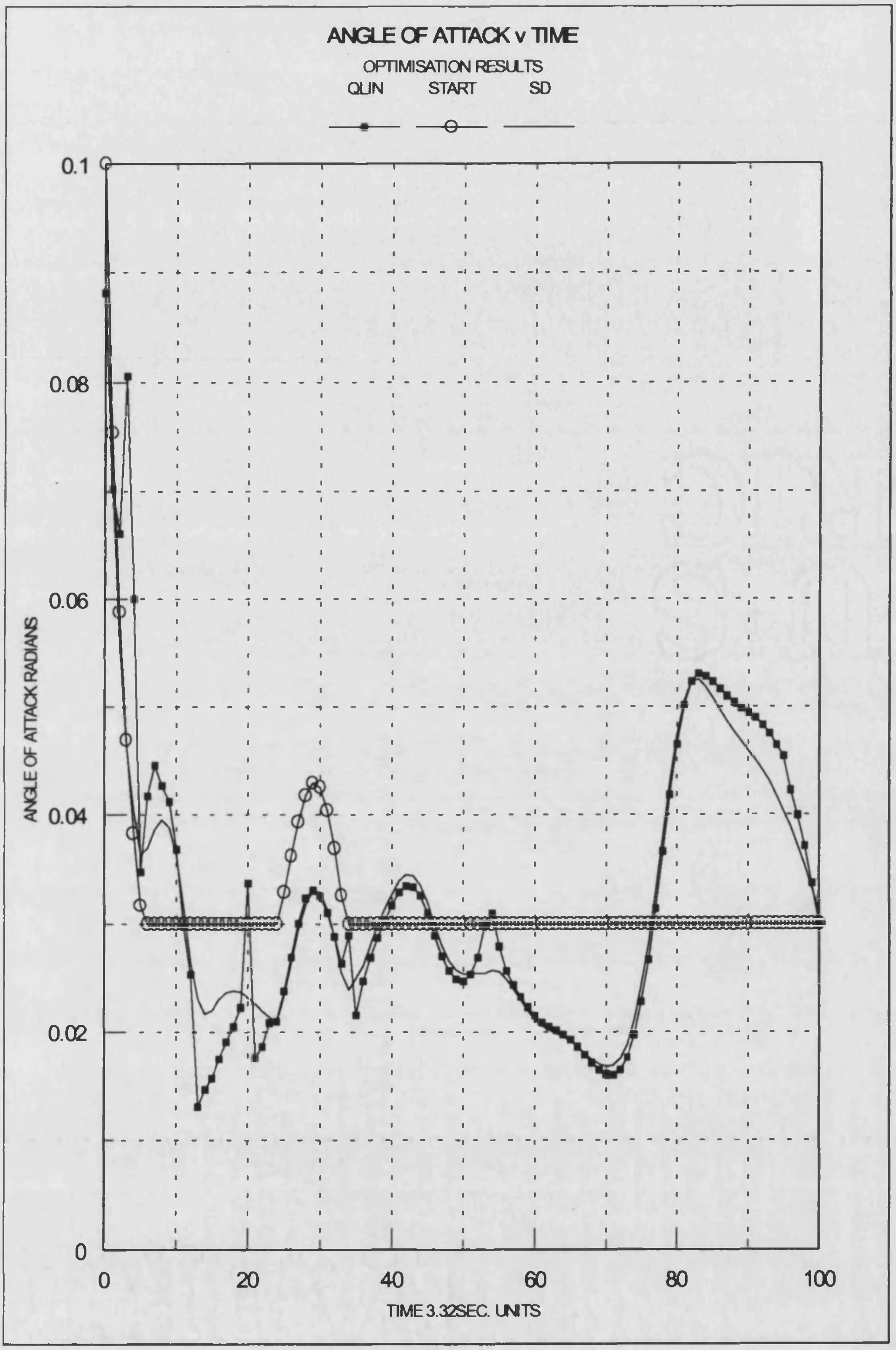

ANGLE OF ATTACK v TIME
OPTIMISATION RESULTS
QLIN
START
SD
0.1
0.08
0.06
0.04
0.02
0
ANGLE OF ATTACK RADIANS
0
20
40
60
80
100
TIME 3.32SEC. UNITS

Fig. 6

Chapter 4

QUASILINEARISATION

In this chapter the method of Quasilinearisation is employed to compute the solution of the two point boundary value problem resulting from the application of the necessary conditions for optimal control. This method of solution requires sets of starting vectors for the state, co-state, and control variables to initiate the iterative computational procedures. The results obtained by the Steepest Descent method of solution described in chapter 3 are used as the starting vectors for the Quasilinearisation procedure described in this chapter.

The Newton-Raphson Algorithm.

A system of non-linear differential equations of the form

$$\dot{\underline{x}} = f(\underline{x},\underline{u},t)$$

can be linearised with respect to the variables x and u and solved by the iterative solution of a linearised set of differential equations as given by:-

$$[\dot{\underline{x}}]_{N+1} = \left[J(\underline{x}_N) \;\vdots\; J(\underline{u}_N)\right]\begin{bmatrix} \underline{x}_{N+1} - \underline{x}_N \\ \hdashline \underline{u}_{N+1} - \underline{u}_N \end{bmatrix} + \left[f(\underline{x}_N, \underline{u}_N)\right]$$

This is the generalised Newton-Raphson Algorithm.

In this algorithm N represents the n^{th} iteration. It is readily seen that as the $(n+1)^{th.}$ iteration converges to the $n^{th.}$, the solution is that of the original non-linear system of equations.

The Jacobian is given by:-

$$[J(\underline{x}_N) \mid J(\underline{u}_N)] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \cdots & \frac{\partial f_1}{\partial x_n} & \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & & & \vdots & \vdots & & \vdots \\ \vdots & & & \vdots & \vdots & & \vdots \\ \vdots & & & \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \cdots & \frac{\partial f_n}{\partial x_n} & \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}_N$$

n is the dimension of the non-linear system of equations.

This Algorithm can be rearranged to give

$$[\dot{\underline{x}}]_{N+1} = [J(\underline{x}_N)][\underline{x}_{N+1}] + \{[J(\underline{u}_N)][\underline{u}_{N+1} - u_N] + [f(\underline{x}_N, u_N)] - [J(\underline{x}_N)][\underline{x}_N]\}$$

This is more easily seen to be of the familiar linear form as given by

$$[\dot{\underline{x}}] = [A][\underline{x}] + [B][\underline{u}]$$

where the Forcing function is

$$\{[J(\underline{u}_N)][\underline{u}_{N+1} - u_N] + [f(\underline{x}_N, u_N)] - [J(\underline{x}_N)][\underline{x}_N]\}$$

and the unforced homogeneous system of equations is

$$[\dot{\underline{x}}]_{N+1} = [J(\underline{x}_N)][\underline{x}_{N+1}]$$

The elements of $[J(\underline{x}_N)]$ are in general non-linear functions of the previous iteration. These elements are now time varying coefficients, and as a result the system is now a linear time varying system.

The solution of such a system of linear differential equations consists of a linear combination of sets of the Homogeneous System solutions plus a Particular Integral. At each iteration the general time solution of the linear system is given by:

$$[\underline{x}(t)]_{N+1} = c_1[\underline{x}_{H_1}(t)] + c_2[\underline{x}_{H2}(t)] + ------ + c_r[\underline{x}_{H_r}(t)] + [\underline{x}_{P.I.}(t)]$$

This solution is valid at all points in time throughout the integration period.
In particular at the terminal time the solution is

$$[\underline{x}(t_f)]_{N+1} = c_1[\underline{x}_{H_1}(t_f)] + c_2[\underline{x}_{H2}(t_f)] + ------ + c_r[\underline{x}_{H_r}(t_f)] + [\underline{x}_{P.I.}(t_f)]$$

Also the solution at the initial time is given by

$$[\underline{x}(t_0)]_{N+1} = c_1[\underline{x}_{H_1}(t_0)] + c_2[\underline{x}_{H2}(t_0)] + ------ + c_r[\underline{x}_{H_r}(t_0)] + [\underline{x}_{P.I.}(t_0)]$$

These facts are applied in the solution of the non-linear boundary value problem resulting from the application of the necessary conditions for optimal control as follows. The complete non-linear set of differential equations to be solved consists of the $\boldsymbol{n}$-dimensional set of system state equations together with a further $\boldsymbol{n}$-dimensional set of co-state equations.
This complete $\mathbf{2}\boldsymbol{n}$-dimensional set of non-linear equations is linearised by applying the Quasilinearisation Algorithm.
In general the initial conditions on the states are known and the terminal conditions on the co-states are also known.
To initiate the iteration process, initial conditions are chosen for the unknown initial conditions of the co-state vector. This enables the integration to be performed forwards in time. A set of so-called starting vectors for each of the state and co-state variables throughout the integration period is also chosen. At each time step in the

integration process the elements of the Jacobian $\left[J(\underline{x}_N)\right]$ are calculated. A Particular Integration of the complete set of differential equations is performed, using the initial conditions of the chosen starting vectors as the initial conditions of the first iteration.

The resultant Terminal conditions obtained from the particular integration $\left[\underline{x}_{P.I.}(t_f)\right]$ will not in general satisfy the required terminal conditions of the co-states., and these must be corrected.

To this end a number of integrations of the Homogeneous system of equations are performed. It is necessary to produce as many sets of homogeneous integrations as there are unknown initial conditions on the co-states. From a knowledge of the desired terminal conditions on the co-states, given by the boundary conditions obtained from the necessary conditions for Optimal Control, it is possible to calculate improved estimates for the initial conditions on the co-states. If these corrected initial conditions are now used to perform a new particular integration, then the terminal values of the co-states will satisfy the desired terminal conditions. The estimates of the unknown initial conditions of the co-states are corrected as follows.

Assuming there are n desired terminal conditions and also n unknown initial conditions on the co-states then the terminal conditions on the co-states are given by

$$\left[\underline{\lambda}_D(t_f)\right]_{N+1} = c_1\left[\underline{\lambda}_{H_1}(t_f)\right] + c_2\left[\underline{\lambda}_{H2}(t_f)\right] + ------ + c_n\left[\underline{\lambda}_{H_n}(t_f)\right] + \left[\underline{\lambda}_{P.I.}(t_f)\right]$$

This equation can be rearranged as

$$\begin{bmatrix} \lambda_{1H_1}(t_f) & \cdots & \cdots & \lambda_{1H_n}(t_f) \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \lambda_{nH_1}(t_f) & \cdots & \cdots & \lambda_{nH_n}(t_f) \end{bmatrix} \begin{bmatrix} c_1 \\ \cdots \\ \cdots \\ c_n \end{bmatrix} = \begin{bmatrix} \lambda_{1D}(t_f) - \lambda_{1P.I.}(t_f) \\ \cdots \\ \cdots \\ \lambda_{nD}(t_f) - \lambda_{nP.I.}(t_f) \end{bmatrix}$$

This equation can be solved for the vector of weighting constants $[\underline{c}]$ provided the matrix consisting of the sets of homogeneous solutions at t_f is non-singular.

$$\begin{bmatrix} c_1 \\ \cdots \\ \cdots \\ c_n \end{bmatrix} = \begin{bmatrix} \lambda_{1H_1}(t_f) & \cdots & \cdots & \lambda_{1H_n}(t_f) \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \lambda_{nH_1}(t_f) & \cdots & \cdots & \lambda_{nH_n}(t_f) \end{bmatrix}^{-1} \begin{bmatrix} \lambda_{1D}(t_f) - \lambda_{1P.I.}(t_f) \\ \cdots \\ \cdots \\ \lambda_{nD}(t_f) - \lambda_{nP.I.}(t_f) \end{bmatrix}$$

These homogeneous weighting constants also apply at time zero and so the corrected initial conditions on the co-states are given by

$$\begin{bmatrix} \lambda_1(t_0) \\ \cdots \\ \cdots \\ \lambda_n(t_0) \end{bmatrix} = \begin{bmatrix} \lambda_{1H_1}(t_0) & \cdots & \cdots & \lambda_{1H_n}(t_0) \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \lambda_{nH_1}(t_0) & \cdots & \cdots & \lambda_{nH_n}(t_0) \end{bmatrix} \begin{bmatrix} c_1 \\ \cdots \\ \cdots \\ c_n \end{bmatrix} + \begin{bmatrix} \lambda_{1P.I.}(t_0) \\ \cdots \\ \cdots \\ \lambda_{nP.I.}(t_0) \end{bmatrix}$$

where the $\left[\underline{\lambda}_{P.I.}(t_0)\right]$ are the original estimates for the unknown initial conditions of the co-states.

If the initial conditions for the sets of homogeneous integrations are specifically chosen to be

$$\left[\begin{array}{c:c:c:c}\underline{x}_{H_1}(t_0) & & & \underline{x}_{H_n}(t_0) \\ \hdashline \underline{\lambda}_{H_1}(t_0) & & & \underline{\lambda}_{H_n}(t_0)\end{array}\right] = \left[\begin{array}{c:c:c:c} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ & & & \\ 0 & 0 & 0 & 0 \\ \hdashline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ & & & \\ 0 & 0 & 0 & 1 \end{array}\right]$$

such that each set of homogeneous initial condition vector has all the elements zero except one of the co-states taken one at a time, then the corrected initial conditions on the co-states become

$$\begin{bmatrix}\lambda_1(t_0) \\ \cdots\cdots \\ \cdots\cdots \\ \cdots\cdots \\ \lambda_n(t_0)\end{bmatrix} = \begin{bmatrix} c_1 \\ \cdots \\ \cdots \\ \cdots \\ c_n\end{bmatrix} + \begin{bmatrix}\lambda_{1P.I.}(t_0) \\ \cdots\cdots \\ \cdots\cdots \\ \cdots\cdots \\ \lambda_{nP.I.}(t_0)\end{bmatrix}$$

A subsequent forward time integration of the complete system of state and co-state equations using the known initial conditions for the states and the corrected initial conditions on the co-states will now satisfy the required terminal conditions on the co-state equations.

This complete integration is the first iteration solution. These time responses for the states and co-states are then used to replace the chosen starting vectors for the second iteration.

This process is repeated with corrections being made to the co-state initial conditions at each iteration and the subsequent N+1 th. solution being used to replace the n th. in the Quasilinearisation Algorithm.

The iteration process continues until convergence occurs as defined by

$$\sum_{i=1}^{1=n}\{x_{i_{N+1}}(t)-x_{i_N}(t)\}^2+\sum_{i=1}^{1=n}\{\lambda_{i_{N+1}}(t)-\lambda_{i_N}(t)\}^2\leq\varepsilon$$

where ε is a suitably chosen small constant.

Bellman has shown (10) that if convergence of the Quasilinearisation iteration process occurs then it is quadratic.

Convergence of the algorithm is unfortunately dependent on a suitable choice of starting vectors. To obtain a suitable choice of starting vectors for both the state and co-states the method of Steepest Descent was employed. This method, although requiring integration of the state equations forwards in time and the integration of the co-states backwards in time, only requires an estimate of the control vector as a function of time to initiate the iteration process. This estimate is simpler to choose, from an engineering knowledge of the process to be optimised, than the estimation of both the state and the co-state trajectories.

Initial convergence of the steepest descent algorithm is not too sensitive to the choice of the control starting vector. Convergence does however become extremely slow as a solution is approached, requiring a very large number of iterations to obtain a solution to the non-linear two point boundary value problem. Attempts to reduce the number of iterations required by increasing the factor tau which controls the displacement along the path of steepest descent can result in instabilities of the computational process.

A combination of both the steepest descent method and the Quasilinearisation algorithm has resulted in a solution to the specific optimisation problem being obtained without computational instabilities occurring. Steepest Descent was used until the convergence rate slowed to an unacceptable level and the results from this method were then used as starting vectors to initiate the Quasilinearisation iteration process as described above.

The steepest descent procedure for the specific optimisation problem of aircraft maximum climb height in a fixed time period with minimum drag is described in detail in chapter 3.

As the correction of the initial condition on the co-states to satisfy the terminal boundary conditions of the co-states requires the inversion of the matrix consisting of the sets of homogeneous solutions at the final time, it is important that this matrix does not become 'Ill-Conditioned'.

To this end a Gram-Schmidt Orthonormalisation was performed every five time steps during the integration of the homogeneous system of differential equations. The orthonormalisation was applied across the full set of homogeneous time solutions resulting from the sets of initial conditions of each homogeneous solution as defined above. The orthonormalisation procedure was applied by using each vector set of homogeneous time solutions after every five integration steps. A total of twenty orthonormalisations were performed over the specified time interval of the optimisation problem and by this means the matrix of homogeneous time solutions at time t_f was prevented from becoming ill-conditioned. The total integration time for the optimisation problem was divided into one hundred equal time steps.

Details of the Gram-Schmidt Orthonormalisation procedure are given below.

Gram-Schmidt Orthonormalisation :-

Define the norm

$$\|G\| = \langle G,G \rangle^{\frac{1}{2}}$$

Then a normalised vector is given by

$$\phi_1 = \frac{F_1(Y)}{\|F_1\|}$$

and in general

$$\phi_k(Y) = \frac{F_k(Y) - \sum_{i=1}^{k-1} \langle F_k, \phi_i \rangle \phi_i(Y)}{\left\| F_k - \sum_{i=1}^{k-1} \langle F_k, \phi_i \rangle \phi_i(Y) \right\|}$$

After each orthonormalisation every five integration steps of the homogeneous sets of solutions, the same transformation was applied to the sets of initial conditions for the homogeneous solutions.

The following section of this chapter applies the Quasilinearisation algorithm to the necessary conditions for the optimisation problem as defined in chapter 2.

Linearisation of the Optimal Control Equation.

Applying the Quasilinearisation Algorithm to the algebraic expression for the Optimal Control.

$$g_1(V,h,m,\lambda_V,\lambda_\gamma,\alpha)=0$$

$$\left[\frac{\partial g_1}{\partial V} \vdots \frac{\partial g_1}{\partial h} \vdots \frac{\partial g_1}{\partial m} \vdots \frac{\partial g_1}{\partial \lambda_V} \vdots \frac{\partial g_1}{\partial \lambda_\gamma} \vdots \frac{\partial g_1}{\partial \alpha}\right]\begin{bmatrix} V_{N+1}-V_N \\ h_{N+1}-h_N \\ m_{N+1}-m_N \\ \lambda_{V\,N+1}-\lambda_{V\,N} \\ \lambda_{\gamma\,N+1}-\lambda_{\gamma\,N} \\ \alpha_{N+1}-\alpha_N \end{bmatrix}+g_1=0$$

From this expression

$$\alpha_{N+1}-\alpha_N=-\frac{1}{\frac{\partial g_1}{\partial \alpha}}\left\{\left[\frac{\partial g_1}{\partial V} \vdots \frac{\partial g_1}{\partial h} \vdots \frac{\partial g_1}{\partial m} \vdots \frac{\partial g_1}{\partial \lambda_V} \vdots \frac{\partial g_1}{\partial \lambda_\gamma}\right]\begin{bmatrix} V_{N+1}-V_N \\ h_{N+1}-h_N \\ m_{N+1}-m_N \\ \lambda_{V\,N+1}-\lambda_{V\,N} \\ \lambda_{\gamma\,N+1}-\lambda_{\gamma\,N} \end{bmatrix}+g_1\right\}$$

$$\frac{\partial g_1}{\partial V}=-\frac{\lambda_V}{m}\{\frac{\partial T}{\partial V}\sin\alpha+\frac{\partial^2 D}{\partial\alpha\partial V}\}+\frac{\lambda_\gamma}{mV^2}\{V(\frac{\partial^2 L}{\partial\alpha\partial V}+\frac{\partial T}{\partial V}\cos\alpha)-(\frac{\partial L}{\partial\alpha}+T\cos\alpha)\}$$

$$\frac{\partial g_1}{\partial h}=-\frac{\lambda_V}{m}\{\frac{\partial T}{\partial h}\sin\alpha+\frac{\partial^2 D}{\partial\alpha\partial h}\}+\frac{\lambda_\gamma}{mV}\{(\frac{\partial^2 L}{\partial\alpha\partial h}+\frac{\partial T}{\partial h}\cos\alpha)\}$$

$$\frac{\partial g_1}{\partial m}=\frac{\lambda_V}{m^2}\{T\sin\alpha+\frac{\partial D}{\partial\alpha}\}-\frac{\lambda_\gamma}{m^2V}\{\frac{\partial L}{\partial\alpha}+T\cos\alpha\}$$

$$\frac{\partial g_1}{\partial \lambda_V} = -\frac{1}{m}\{T\sin\alpha + \frac{\partial D}{\partial \alpha}\}$$

$$\frac{\partial g_1}{\partial \lambda_\gamma} = \frac{1}{mV}\{\frac{\partial L}{\partial \alpha} + T\cos\alpha\}$$

$$\frac{\partial g_1}{\partial \alpha} = 1 - \frac{\lambda_V}{m}\{T\cos\alpha + \frac{\partial^2 D}{\partial \alpha^2}\} + \frac{\lambda_\gamma}{mV}\{\frac{\partial^2 L}{\partial \alpha^2} - T\sin\alpha\}$$

Linearisation of the State and Costate Equations.

$$J(x_N, u_N) = \begin{bmatrix}
\frac{\partial f_1}{\partial V} & \frac{\partial f_1}{\partial \gamma} & \frac{\partial f_1}{\partial h} & 0 & \frac{\partial f_1}{\partial m} & 0 & 0 & 0 & 0 & 0 & \frac{\partial f_1}{\partial \alpha} \\
\frac{\partial f_2}{\partial V} & \frac{\partial f_2}{\partial \gamma} & \frac{\partial f_2}{\partial h} & 0 & \frac{\partial f_2}{\partial m} & 0 & 0 & 0 & 0 & 0 & \frac{\partial f_2}{\partial \alpha} \\
\frac{\partial f_3}{\partial V} & \frac{\partial f_3}{\partial \gamma} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial f_4}{\partial V} & \frac{\partial f_4}{\partial \gamma} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial f_5}{\partial V} & 0 & \frac{\partial f_5}{\partial h} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial f_6}{\partial V} & \frac{\partial f_6}{\partial \gamma} & \frac{\partial f_6}{\partial h} & 0 & \frac{\partial f_6}{\partial m} & \frac{\partial f_6}{\partial \lambda_V} & \frac{\partial f_6}{\partial \lambda_\gamma} & \frac{\partial f_6}{\partial \lambda_h} & \frac{\partial f_6}{\partial \lambda_x} & \frac{\partial f_6}{\partial \lambda_m} & \frac{\partial f_6}{\partial \alpha} \\
\frac{\partial f_7}{\partial V} & \frac{\partial f_7}{\partial \gamma} & 0 & 0 & 0 & \frac{\partial f_7}{\partial \lambda_V} & \frac{\partial f_7}{\partial \lambda_\gamma} & \frac{\partial f_7}{\partial \lambda_h} & \frac{\partial f_7}{\partial \lambda_x} & 0 & 0 \\
\frac{\partial f_8}{\partial V} & 0 & \frac{\partial f_8}{\partial h} & 0 & \frac{\partial f_8}{\partial m} & \frac{\partial f_8}{\partial \lambda_V} & \frac{\partial f_8}{\partial \lambda_\gamma} & 0 & 0 & \frac{\partial f_8}{\partial \lambda_m} & \frac{\partial f_8}{\partial \alpha} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial f_{10}}{\partial V} & 0 & \frac{\partial f_{10}}{\partial h} & 0 & \frac{\partial f_{10}}{\partial m} & \frac{\partial f_{10}}{\partial \lambda_V} & \frac{\partial f_{10}}{\partial \lambda_\gamma} & 0 & 0 & 0 & \frac{\partial f_{10}}{\partial \alpha}
\end{bmatrix}_N$$

In the linearisation of the State and Costate equations it is desirable that the control appears only as a function of the previous iteration N and not as an explicit function of the current iteration N+1. To this end the expression for $\alpha_{N+1} - \alpha_N$, obtained from the linearisation of the optimal control function, is substituted into the linearised State and Costate equations. Since the incremental control variable $\alpha_{N+1} - \alpha_N$ is a function of the incremental states and co-states, this substitution modifies the elements of the linearised system matrix as well as the forcing function portion of the system of state and co-state equations.

The Jacobian $\left[J(\underline{x}_N)\right]$ then becomes

$$
\left[\begin{array}{c:c:c:c:c:c:c:c:c:c}
\frac{\partial f_1}{\partial V} - a_1\frac{\partial g_1}{\partial V} & \frac{\partial f_1}{\partial \gamma} & \frac{\partial f_1}{\partial h} - a_1\frac{\partial g_1}{\partial h} & 0 & \frac{\partial f_1}{\partial m} - a_1\frac{\partial g_1}{\partial m} & -a_1\frac{\partial g_1}{\partial \lambda_V} & -a_1\frac{\partial g_1}{\partial \lambda_\gamma} & 0 & 0 & 0 \\ \hdashline
\frac{\partial f_2}{\partial V} - a_2\frac{\partial g_1}{\partial V} & \frac{\partial f_2}{\partial \gamma} & \frac{\partial f_2}{\partial h} - a_2\frac{\partial g_1}{\partial h} & 0 & \frac{\partial f_2}{\partial m} - a_2\frac{\partial g_1}{\partial m} & -a_2\frac{\partial g_1}{\partial \lambda_V} & -a_2\frac{\partial g_1}{\partial \lambda_\gamma} & 0 & 0 & 0 \\ \hdashline
\frac{\partial f_3}{\partial V} & \frac{\partial f_3}{\partial \gamma} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline
\frac{\partial f_4}{\partial V} & \frac{\partial f_4}{\partial \gamma} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline
\frac{\partial f_5}{\partial V} & 0 & \frac{\partial f_5}{\partial h} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline
\frac{\partial f_6}{\partial V} - a_6\frac{\partial g_1}{\partial V} & \frac{\partial f_6}{\partial \gamma} & \frac{\partial f_6}{\partial h} - a_6\frac{\partial g_1}{\partial h} & 0 & \frac{\partial f_6}{\partial m} - a_6\frac{\partial g_1}{\partial m} & \frac{\partial f_6}{\partial \lambda_V} - a_6\frac{\partial g_1}{\partial \lambda_V} & \frac{\partial f_6}{\partial \lambda_\gamma} - a_6\frac{\partial g_1}{\partial \lambda_\gamma} & \frac{\partial f_6}{\partial \lambda_h} & \frac{\partial f_6}{\partial \lambda_x} & \frac{\partial f_6}{\partial \lambda_m} \\ \hdashline
\frac{\partial f_7}{\partial V} & \frac{\partial f_7}{\partial \gamma} & 0 & 0 & 0 & \frac{\partial f_7}{\partial \lambda_V} & \frac{\partial f_7}{\partial \lambda_\gamma} & \frac{\partial f_7}{\partial \lambda_h} & \frac{\partial f_7}{\partial \lambda_x} & 0 \\ \hdashline
\frac{\partial f_8}{\partial V} - a_8\frac{\partial g_1}{\partial V} & 0 & \frac{\partial f_8}{\partial h} - a_8\frac{\partial g_1}{\partial h} & 0 & \frac{\partial f_8}{\partial m} - a_8\frac{\partial g_1}{\partial m} & \frac{\partial f_8}{\partial \lambda_V} - a_8\frac{\partial g_1}{\partial \lambda_V} & \frac{\partial f_8}{\partial \lambda_\gamma} - a_8\frac{\partial g_1}{\partial \lambda_\gamma} & 0 & 0 & \frac{\partial f_8}{\partial \lambda_m} \\ \hdashline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline
\frac{\partial f_{10}}{\partial V} - a_{10}\frac{\partial g_1}{\partial V} & 0 & \frac{\partial f_{10}}{\partial h} - a_{10}\frac{\partial g_1}{\partial h} & 0 & \frac{\partial f_{10}}{\partial m} - a_{10}\frac{\partial g_1}{\partial m} & \frac{\partial f_{10}}{\partial \lambda_V} - a_{10}\frac{\partial g_1}{\partial \lambda_V} & \frac{\partial f_{10}}{\partial \lambda_\gamma} - a_{10}\frac{\partial g_1}{\partial \lambda_\gamma} & 0 & 0 & 0
\end{array}\right]_N
$$

$$\text{where } a_i = \frac{\partial f_i}{\partial \alpha} \div \frac{\partial g_1}{\partial \alpha}$$

and the partial derivatives are as defined below:-

$$\frac{\partial f_1}{\partial \alpha} = -\frac{1}{m}\left(T\sin\alpha + \frac{\partial D}{\partial \alpha}\right)$$

$$\frac{\partial f_2}{\partial \alpha} = \frac{1}{mV}\left(\frac{\partial L}{\partial \alpha} + T\cos\alpha\right)$$

$$\frac{\partial f_6}{\partial \alpha}=\frac{\lambda_V}{m}(\frac{\partial^2 D}{\partial V\partial \alpha}+\frac{\partial T}{\partial V}\sin\alpha)+\frac{\lambda_\gamma}{mV^2}\{(\frac{\partial L}{\partial \alpha}+T\cos\alpha)-V(\frac{\partial T}{\partial V}\cos\alpha+\frac{\partial^2 L}{\partial V\partial \alpha})\}$$

$$\frac{\partial f_8}{\partial \alpha}=\frac{\lambda_V}{m}(\frac{\partial^2 D}{\partial h\partial \alpha}+\frac{\partial T}{\partial h}\sin\alpha)-\frac{\lambda_\gamma}{mV}(\frac{\partial T}{\partial h}\cos\alpha+\frac{\partial^2 L}{\partial h\partial \alpha})$$

$$\frac{\partial f_{10}}{\partial \alpha}=-\frac{\lambda_V}{m^2}(T\sin\alpha+\frac{\partial D}{\partial \alpha})+\frac{\lambda_\gamma}{m^2V}(\frac{\partial L}{\partial \alpha}+T\cos\alpha)$$

$$\frac{\partial f_6}{\partial V}=\frac{\lambda_V}{m}(\frac{\partial^2 D}{\partial V^2}-\frac{\partial^2 T}{\partial V^2}\cos\alpha)-\frac{2\lambda_\gamma}{mV^2}\{\frac{1}{V}(L+T\sin\alpha-mg\cos\gamma)$$
$$-(\frac{\partial L}{\partial V}+\frac{\partial T}{\partial V}\sin\alpha)+\frac{V}{2}(\frac{\partial^2 T}{\partial V^2}\sin\alpha+\frac{\partial^2 L}{\partial V^2})\}+\frac{\lambda_m}{cg}\frac{\partial^2 T}{\partial V^2}$$

$$\frac{\partial f_7}{\partial V}=\lambda_\gamma\frac{g}{V^2}\sin\gamma-\lambda_h\cos\gamma+\lambda_x\sin\gamma$$

$$\frac{\partial f_8}{\partial V}=\frac{\lambda_V}{m}(\frac{\partial^2 D}{\partial h\partial V}-\frac{\partial^2 T}{\partial h\partial V}\cos\alpha)-\frac{\lambda_\gamma}{mV^2}\{V(\frac{\partial^2 T}{\partial h\partial V}\sin\alpha+\frac{\partial^2 L}{\partial h\partial V})$$
$$-(\frac{\partial T}{\partial h}\sin\alpha+\frac{\partial L}{\partial h})\}+\frac{\lambda_m}{cg}\frac{\partial^2 T}{\partial h\partial V}$$

$$\frac{\partial f_{10}}{\partial V}=\frac{\lambda_V}{m^2}(\frac{\partial T}{\partial V}\cos\alpha-\frac{\partial D}{\partial V})+\frac{\lambda_\gamma}{m^2V^2}\{V(\frac{\partial L}{\partial V}+\frac{\partial T}{\partial V}\sin\alpha)-(L+T\sin\alpha)\}$$

$$\frac{\partial f_6}{\partial \gamma}=\frac{\lambda_\gamma}{V^2}(g\sin\gamma)-\lambda_h\cos\gamma+\lambda_x\sin\gamma$$

$$\frac{\partial f_7}{\partial \gamma}=-\lambda_V g\sin\gamma-\lambda_\gamma\frac{g}{V}\cos\gamma+\lambda_h V\sin\gamma+\lambda_x V\cos\gamma$$

$$\frac{\partial f_6}{\partial h}=\frac{\lambda_V}{m}(\frac{\partial^2 D}{\partial V\partial h}-\frac{\partial^2 T}{\partial V\partial h}\cos\alpha)+\frac{\lambda_\gamma}{mV^2}\{(\frac{\partial L}{\partial h}+\frac{\partial T}{\partial h}\sin\alpha)$$
$$-V(\frac{\partial^2 T}{\partial V\partial h}\sin\alpha+\frac{\partial^2 L}{\partial V\partial h})\}+\frac{\lambda_m}{cg}\frac{\partial^2 T}{\partial V\partial h}$$

$$\frac{\partial f_8}{\partial h}=\frac{\lambda_V}{m}(\frac{\partial^2 D}{\partial h^2}-\frac{\partial^2 T}{\partial h^2}\cos\alpha)-\frac{\lambda_\gamma}{mV}(\frac{\partial^2 T}{\partial h^2}\sin\alpha+\frac{\partial^2 L}{\partial h^2})+\frac{\lambda_m}{cg}\frac{\partial^2 T}{\partial h^2}$$

$$\frac{\partial f_{10}}{\partial h}=\frac{\lambda_V}{m^2}(\frac{\partial T}{\partial h}\cos\alpha-\frac{\partial D}{\partial h})+\frac{\lambda_\gamma}{m^2V}(\frac{\partial L}{\partial h}+\frac{\partial T}{\partial h}\sin\alpha)$$

$$\frac{\partial f_6}{\partial m}=\frac{\lambda_V}{m^2}(\frac{\partial T}{\partial V}\cos\alpha-\frac{\partial D}{\partial V})+\frac{\lambda_\gamma}{m^2V^2}\{V(\frac{\partial T}{\partial V}\sin\alpha+\frac{\partial L}{\partial V})-(L+T\sin\alpha)\}$$

$$\frac{\partial f_8}{\partial m}=\frac{\lambda_V}{m^2}(\frac{\partial T}{\partial h}\cos\alpha-\frac{\partial D}{\partial h})+\frac{\lambda_\gamma}{m^2V}(\frac{\partial T}{\partial h}\sin\alpha+\frac{\partial L}{\partial h})$$

$$\frac{\partial f_{10}}{\partial m}=\frac{2\lambda_V}{m^3}(D-T\cos\alpha)-\frac{2\lambda_\gamma}{m^3V}(L+T\sin\alpha)$$

$$\frac{\partial f_6}{\partial \lambda_V}=\frac{1}{m}(\frac{\partial D}{\partial V}-\frac{\partial T}{\partial V}\cos\alpha)$$

$$\frac{\partial f_7}{\partial \lambda_V}=g\cos\gamma$$

$$\frac{\partial f_8}{\partial \lambda_V}=\frac{1}{m}(\frac{\partial D}{\partial h}-\frac{\partial T}{\partial h}\cos\alpha)$$

$$\frac{\partial f_{10}}{\partial \lambda_V}=\frac{1}{m^2}(T\cos\alpha-D)$$

$$\frac{\partial f_6}{\partial \lambda_\gamma}=\frac{1}{mV^2}\{(L+T\sin\alpha-mg\cos\gamma)-V(\frac{\partial T}{\partial V}\sin\alpha+\frac{\partial L}{\partial V})\}$$

$$\frac{\partial f_7}{\partial \lambda_\gamma} = -\frac{g}{V}\sin\gamma$$

$$\frac{\partial f_8}{\partial \lambda_\gamma} = -\frac{1}{mV}\left(\frac{\partial T}{\partial h}\sin\alpha + \frac{\partial L}{\partial h}\right)$$

$$\frac{\partial f_{10}}{\partial \lambda_\gamma} = \frac{1}{m^2 V}(L + T\sin\alpha)$$

$$\frac{\partial f_6}{\partial \lambda_h} = -\sin\gamma$$

$$\frac{\partial f_7}{\partial \lambda_h} = -V\cos\gamma$$

$$\frac{\partial f_6}{\partial \lambda_x} = -\cos\gamma$$

$$\frac{\partial f_7}{\partial \lambda_x} = V\sin\gamma$$

$$\frac{\partial f_6}{\partial \lambda_m} = \frac{1}{cg}\frac{\partial T}{\partial V}$$

$$\frac{\partial f_8}{\partial \lambda_m} = \frac{1}{cg}\frac{\partial T}{\partial h}$$

The Linearised Forcing Function:-

$$f(\underline{x}_N, \underline{u}_N) - J(\underline{x}_N)\underline{x}_N - a_i g_1$$

The Partial Derivatives:-

$$\frac{\partial L}{\partial V}=\rho VSC_{L_\alpha}\alpha+\frac{1}{2}\rho V^2 S\frac{\partial C_{L_\alpha}}{\partial V}\alpha$$

$$\frac{\partial C_{L_\alpha}}{\partial V}=\frac{\partial M}{\partial V}\frac{\partial C_{L_\alpha}}{\partial M}=\frac{1}{a}\frac{\partial C_{L_\alpha}}{\partial M}$$

$$\therefore\frac{\partial L}{\partial V}=\rho VSC_{L_\alpha}\alpha+\frac{1}{2}\rho VSM\frac{\partial C_{L_\alpha}}{\partial M}\alpha$$

$$\frac{\partial L}{\partial V}=\rho VS\alpha(C_{L_\alpha}+\frac{1}{2}M\frac{\partial C_{L_\alpha}}{\partial M})$$

$$\frac{\partial L}{\partial h}=\frac{1}{2}\frac{\partial\rho}{\partial h}V^2SC_{L_\alpha}\alpha=-\frac{1}{2}\frac{\rho}{h_1}V^2SC_{L_\alpha}\alpha=-\frac{L}{h_1}$$

$$\frac{\partial L}{\partial\alpha}=\frac{1}{2}\rho V^2SC_{L_\alpha}$$

$$\frac{\partial^2 L}{\partial V^2}=\rho S\alpha(C_{L_\alpha}+\frac{1}{2}M\frac{\partial C_{L_\alpha}}{\partial M})+\rho VS\alpha(\frac{\partial C_{L_\alpha}}{\partial V}+\frac{1}{2}\frac{\partial M}{\partial V}\frac{\partial C_{L_\alpha}}{\partial M}+\frac{1}{2}M\frac{\partial^2 C_{L_\alpha}}{\partial V\partial M})$$

$$\frac{\partial^2 L}{\partial V^2}=\rho S\alpha(C_{L_\alpha}+\frac{1}{2}M\frac{\partial C_{L_\alpha}}{\partial M})+\rho VS\alpha(\frac{\partial M}{\partial V}\frac{\partial C_{L_\alpha}}{\partial M}+\frac{1}{2}\frac{\partial M}{\partial V}\frac{\partial C_{L_\alpha}}{\partial M}+\frac{1}{2}M\frac{\partial M}{\partial V}\frac{\partial^2 C_{L_\alpha}}{\partial^2 M})$$

$$\frac{\partial^2 L}{\partial V^2}=\rho S\alpha(C_{L_\alpha}+2M\frac{\partial C_{L_\alpha}}{\partial M}+\frac{1}{2}M^2\frac{\partial^2 C_{L_\alpha}}{\partial^2 M})\approx\rho S\alpha(C_{L_\alpha}+2M\frac{\partial C_{L_\alpha}}{\partial M})$$

$$\frac{\partial^2 L}{\partial h\partial V}=-\frac{\rho}{h_1}\rho VS\alpha(C_{L_\alpha}+\frac{1}{2}M\frac{\partial C_{L_\alpha}}{\partial M})=-\frac{1}{h_1}\frac{\partial L}{\partial V}$$

$$\frac{\partial^2 L}{\partial\alpha\partial V}=\rho VSC_{L_\alpha}+\frac{1}{2}\rho V^2S\frac{\partial C_{L_\alpha}}{\partial V}=\rho VSC_{L_\alpha}+\frac{1}{2}\rho V^2S\frac{\partial M}{\partial V}\frac{\partial C_{L_\alpha}}{\partial M}$$

$$\frac{\partial^2 L}{\partial\alpha\partial V}=\rho VS(C_{L_\alpha}+\frac{1}{2}M\frac{\partial C_{L_\alpha}}{\partial M})$$

$$\frac{\partial^2 L}{\partial h^2}=\frac{1}{2}\frac{\partial^2\rho}{\partial h^2}V^2SC_{L_\alpha}\alpha=\frac{1}{2}\frac{\rho}{h_1^{\,2}}V^2SC_{L_\alpha}\alpha=\frac{L}{h_1^{\,2}}$$

$$\frac{\partial^2 L}{\partial\alpha\partial h}=-\frac{1}{2}\frac{\rho}{h_1}V^2SC_{L_\alpha}=-\frac{1}{h_1}\frac{\partial L}{\partial\alpha}$$

$$\frac{\partial^2 L}{\partial\alpha^2}=0$$

$$\frac{\partial D}{\partial V}=\rho VS(C_{D_0}+\eta C_{L_\alpha}\alpha^2)+\frac{1}{2}\rho V^2S(\frac{\partial C_{D_0}}{\partial V}+\frac{\partial\eta C_{L_\alpha}}{\partial V}\alpha^2)$$

$$\frac{\partial C_{D_0}}{\partial V}=\frac{\partial M}{\partial V}\frac{\partial C_{D_0}}{\partial M}=\frac{1}{a}\frac{\partial C_{D_0}}{\partial M}$$

$$\frac{\partial\eta C_{L_\alpha}}{\partial V}=\frac{\partial M}{\partial V}\frac{\partial\eta C_{L_\alpha}}{\partial M}=\frac{1}{a}\frac{\partial\eta C_{L_\alpha}}{\partial M}$$

$$\therefore\frac{\partial D}{\partial V}=\rho VS(C_{D_0}+\eta C_{L_\alpha}\alpha^2)+\frac{1}{2}\rho VSM(\frac{\partial C_{D_0}}{\partial M}+\frac{\partial\eta C_{L_\alpha}}{\partial M}\alpha^2)$$

$$\frac{\partial D}{\partial V}=\rho VS\{(C_{D_0}+\frac{M}{2}\frac{\partial C_{D_0}}{\partial M})+(\eta C_{L_\alpha}+\frac{M}{2}\frac{\partial\eta C_{L_\alpha}}{\partial M})\alpha^2\}$$

$$\frac{\partial D}{\partial h}=\frac{1}{2}\frac{\partial\rho}{\partial h}V^2S(C_{D_0}+\eta C_{L_\alpha}\alpha^2)=-\frac{1}{2}\frac{\rho}{h_1}V^2S(C_{D_0}+\eta C_{L_\alpha}\alpha^2)=-\frac{D}{h_1}$$

$$\frac{\partial D}{\partial\alpha}=\rho V^2S\eta C_{L_\alpha}\alpha$$

Subroutine Aero computes $\frac{\partial C_{L_\alpha}}{\partial M},\frac{\partial C_{D_0}}{\partial M}$ and $\frac{\partial\eta C_{L_\alpha}}{\partial M}$.

$$\frac{\partial T}{\partial V} = \frac{\partial M}{\partial V}\frac{\partial T}{\partial M} = \frac{1}{a}\frac{\partial T}{\partial M}$$

$\frac{\partial T}{\partial M}$ is interpolated from the slope of the Thrust v Mach curve at the appropriate height. Fig (4) refers.

$\frac{\partial T}{\partial h}$ is interpolated from the slope of the Thrust v Height curve at the appropriate Mach number. Fig (4) refers.

Subroutine Thrust computes $\frac{\partial T}{\partial h}$ and $\frac{\partial T}{\partial M}$.

The second derivatives of Thrust are taken as

$$\frac{\partial^2 T}{\partial h^2}=-\frac{1}{h_1}\frac{\partial T}{\partial h}$$

$$\frac{\partial^2 T}{\partial V\partial h}=-\frac{1}{h_1}\frac{\partial T}{\partial V}$$

$$\frac{\partial^2 T}{\partial V^2}=0$$

$$\frac{\partial^2 D}{\partial V^2}=\rho S\{(C_{D_0}+\frac{M}{2}\frac{\partial C_{D_0}}{\partial M})+(\eta C_{L_\alpha}+\frac{M}{2}\frac{\partial \eta C_{L_\alpha}}{\partial M})\alpha^2\}+\rho VS\{(\frac{\partial C_{D_0}}{\partial V}+\frac{\partial \eta C_{L_\alpha}}{\partial V}\alpha^2)$$
$$+\frac{1}{2}\frac{\partial M}{\partial V}(\frac{\partial C_{D_0}}{\partial M}+\frac{\partial \eta C_{L_\alpha}}{\partial M}\alpha^2)+\frac{M}{2}\frac{\partial M}{\partial V}(\frac{\partial^2 C_{D_0}}{\partial M^2}+\frac{\partial^2 \eta C_{L_\alpha}}{\partial M^2}\alpha^2)\}$$

$$\frac{\partial^2 D}{\partial V^2}=\rho S\{(C_{D_0}+\frac{M}{2}\frac{\partial C_{D_0}}{\partial M})+(\eta C_{L_\alpha}+\frac{M}{2}\frac{\partial \eta C_{L_\alpha}}{\partial M})\alpha^2\}+\rho MS\{(\frac{\partial C_{D_0}}{\partial M}+\frac{\partial \eta C_{L_\alpha}}{\partial M}\alpha^2)$$
$$+\frac{1}{2}(\frac{\partial C_{D_0}}{\partial M}+\frac{\partial \eta C_{L_\alpha}}{\partial M}\alpha^2)+\frac{M}{2}(\frac{\partial^2 C_{D_0}}{\partial M^2}+\frac{\partial^2 \eta C_{L_\alpha}}{\partial M^2}\alpha^2)\}$$

$$\frac{\partial^2 D}{\partial V^2}=\rho S\{(C_{D_0}+2M\frac{\partial C_{D_0}}{\partial M})+(\eta C_{L_\alpha}+2M\frac{\partial \eta C_{L_\alpha}}{\partial M})\alpha^2+\frac{M^2}{2}(\frac{\partial^2 C_{D_0}}{\partial M^2}+\frac{\partial^2 \eta C_{L_\alpha}}{\partial M^2}\alpha^2)\}$$

$$\frac{\partial^2 D}{\partial V^2}\approx\rho S\{(C_{D_0}+\eta C_{L_\alpha}\alpha^2)+2M(\frac{\partial C_{D_0}}{\partial M}+\frac{\partial \eta C_{L_\alpha}}{\partial M}\alpha^2)\}$$

$$\frac{\partial^2 D}{\partial V\partial h}=-\frac{\rho}{h_1}VS\{(C_{D_0}+\frac{M}{2}\frac{\partial C_{D_0}}{\partial M})+(\eta C_{L_\alpha}+\frac{M}{2}\frac{\partial \eta C_{L_\alpha}}{\partial M})\alpha^2\}=-\frac{1}{h_1}\frac{\partial D}{\partial V}$$

$$\frac{\partial^2 D}{\partial V\partial \alpha}=\rho VS(2\eta C_{L_\alpha}+M\frac{\partial \eta C_{L_\alpha}}{\partial M})\alpha$$

$$\frac{\partial^2 D}{\partial h^2}=\frac{1}{2}\frac{\partial^2 \rho}{\partial h^2}V^2S(C_{D_0}+\eta C_{L_\alpha}\alpha^2)=\frac{1}{2}\frac{\rho}{h_1^2}V^2S(C_{D_0}+\eta C_{L_\alpha}\alpha^2)=\frac{D}{h_1^2}$$

$$\frac{\partial^2 D}{\partial h \partial \alpha} = -\frac{1}{2}\frac{\rho}{h_1} V^2 S \eta C_{L_\alpha} \alpha$$

$$\frac{\partial^2 D}{\partial \alpha^2} = \rho V^2 S \eta C_{L_\alpha}$$

The results obtained by the Quasilinearisation method of solution of the two point boundary value problem are shown in figs. 7-24. These are presented as time histories of the state and co-state variables during the optimisation interval. It can be seen that the specified terminal constraint on velocity corresponding to Mach 1 at the terminal time is exactly satisfied.

The optimum solution for the maximum height using the defined data is about 70000ft.

Also shown are the optimal time responses for the control α together with the computed Thrust time history. The Optimum results for pitch attitude, pitch rate, normal acceleration and dynamic pressure are included for completeness.

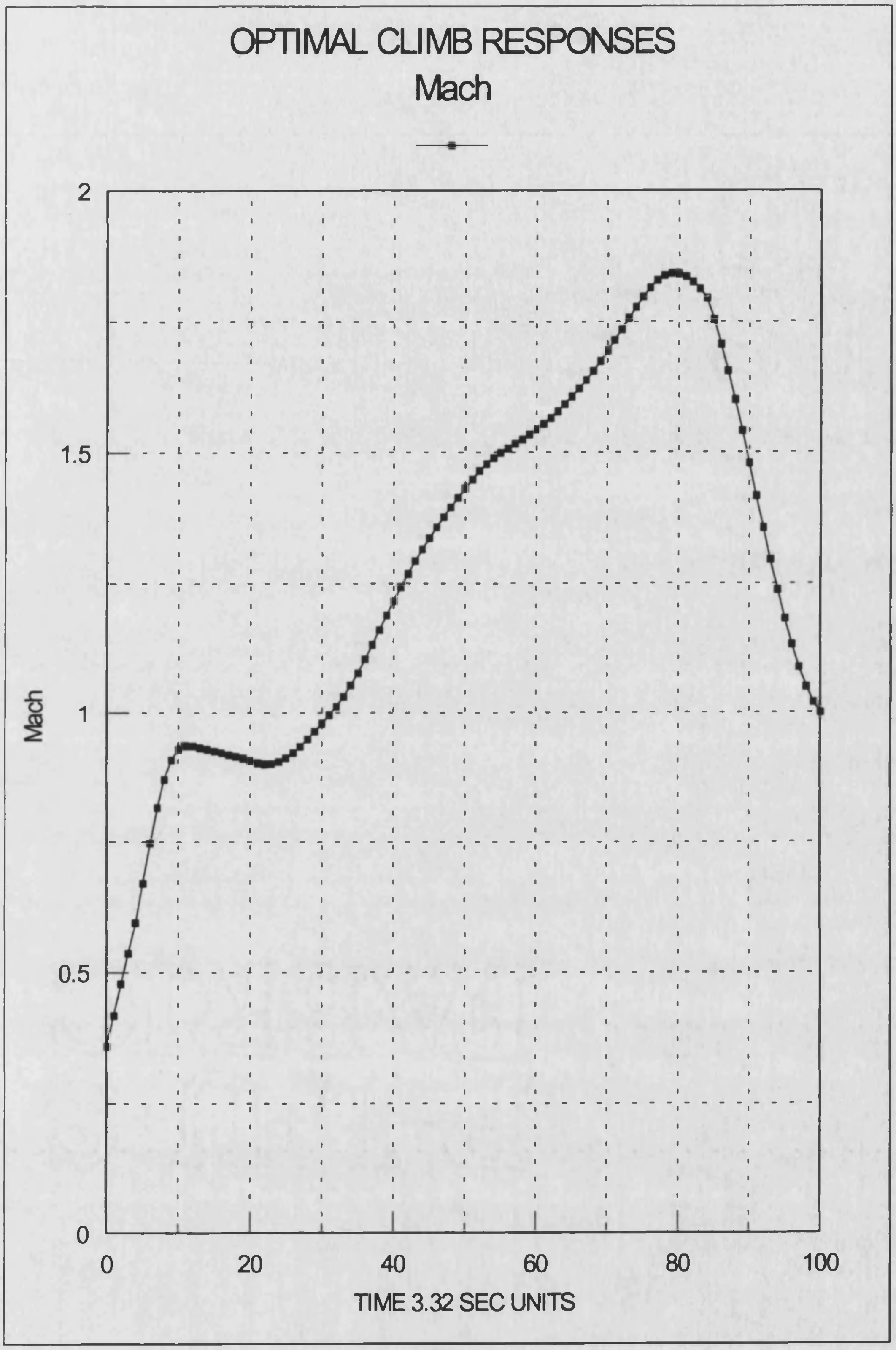
Since the relationship between flight-path angle, pitch attitude and angle of incidence is given by

$$\gamma = \theta - \alpha$$

it is a simple matter to compute the optimum pitch attitude trajectory from the optimum results for flight-path angle and incidence.

Differentiation of the pitch attitude signal with respect to time provides the nominal optimum pitch rate signal $q^*(t)$ which if achieved on the aircraft will produce the optimal climb profile. $q^*(t)$ is to be used as the excitation for a closed pitch rate loop consisting of a command stability augmentation system in series with the aircraft dynamics, in order to fly the optimum manoeuvre.

The following chapter investigates the variation in aircraft short period response characteristics encountered during the optimal climb trajectory. It is this variation in aircraft response characteristics which results in the need for an adaptive C.S.A.S. in order to achieve a uniform pitch rate response at all flight conditions. This then will actually achieve the nominal optimal pitch rate as computed for the optimal climb manoeuvre in this chapter. A listing of the Quasilinearisation programme written for and used in the solution of the optimal climb problem is presented in Appendix C.



OPTIMAL CLIMB RESPONSES
Mach
2
1.5
1
0.5
0
Mach
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 7

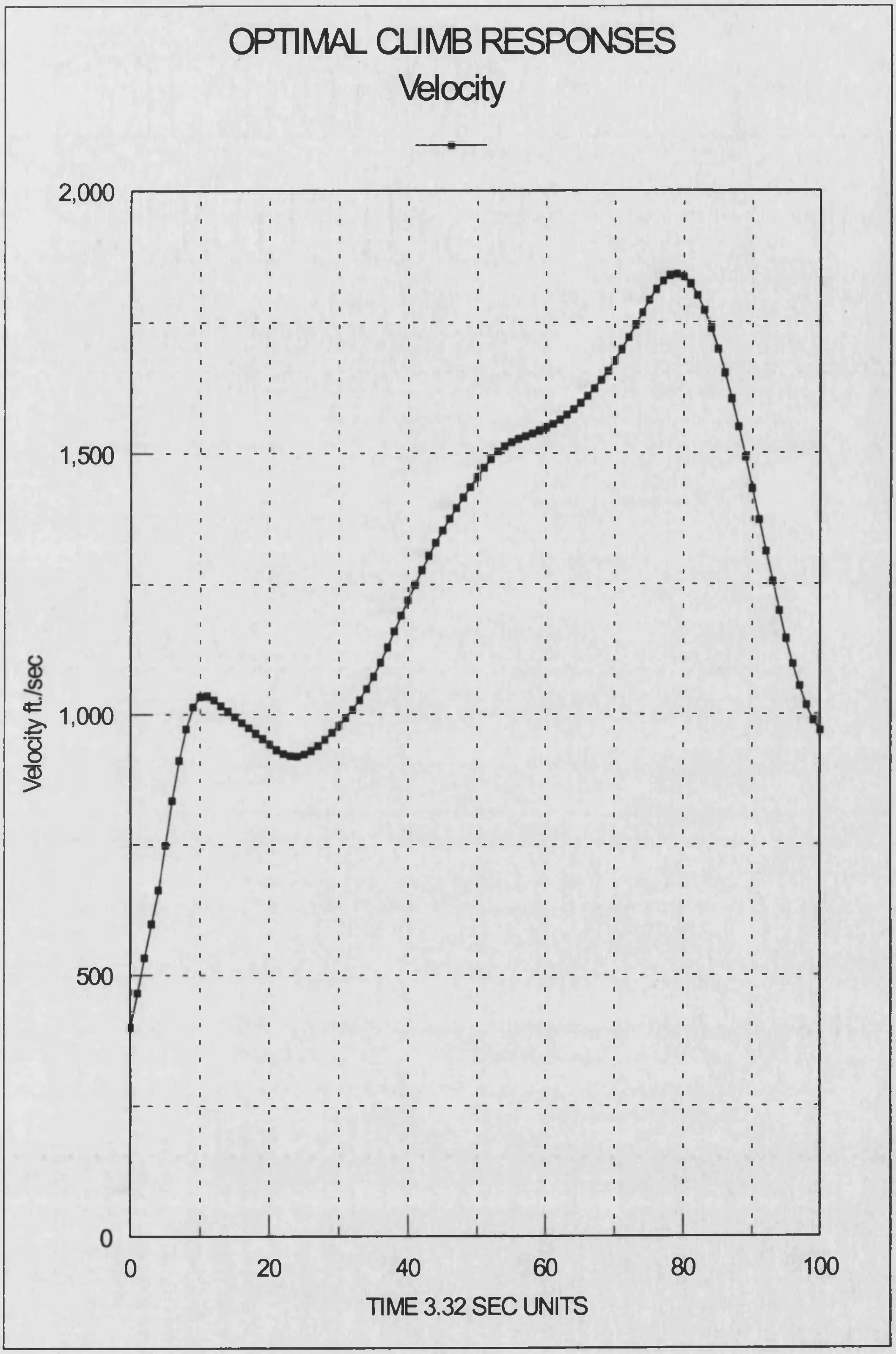
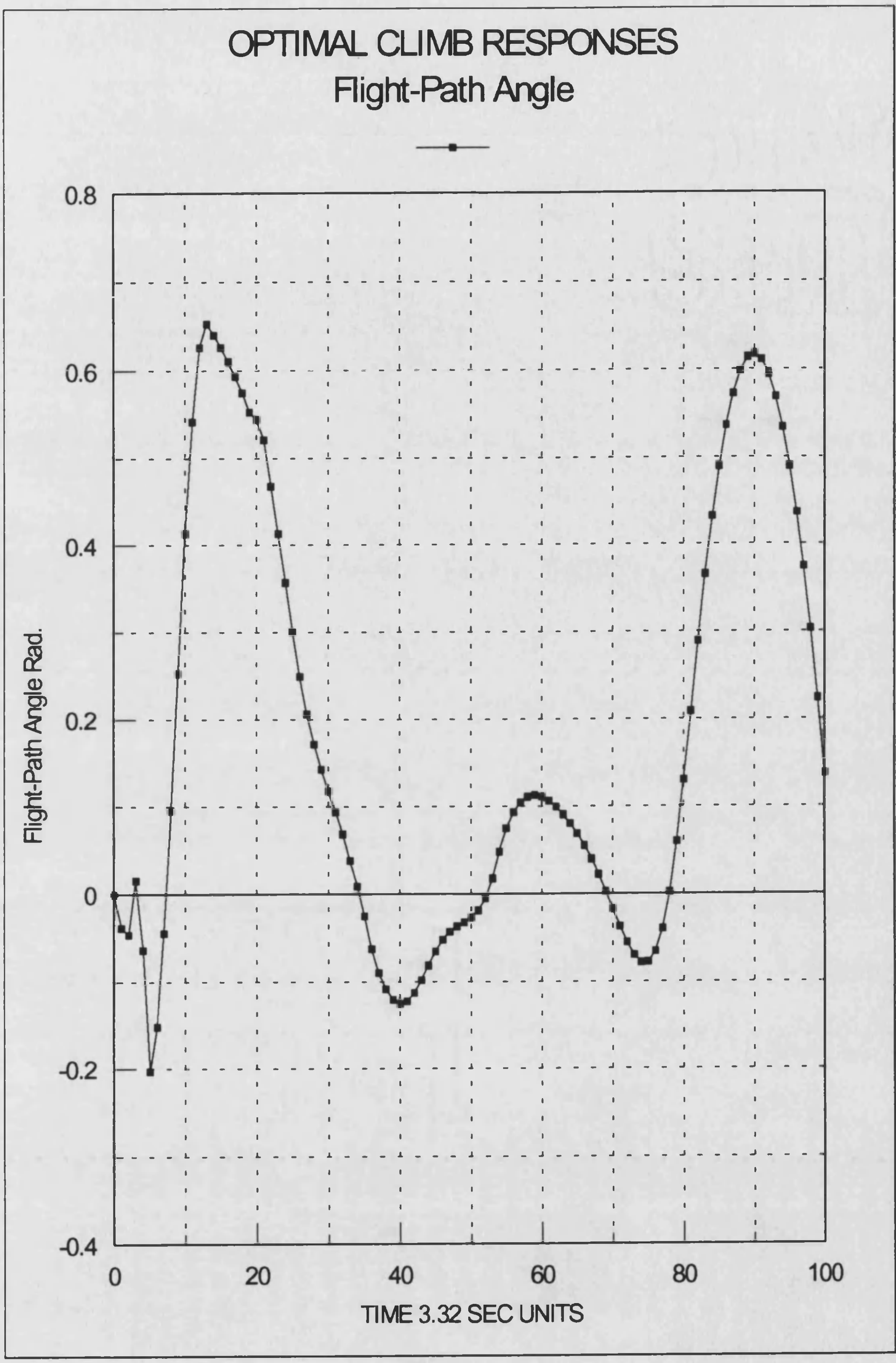
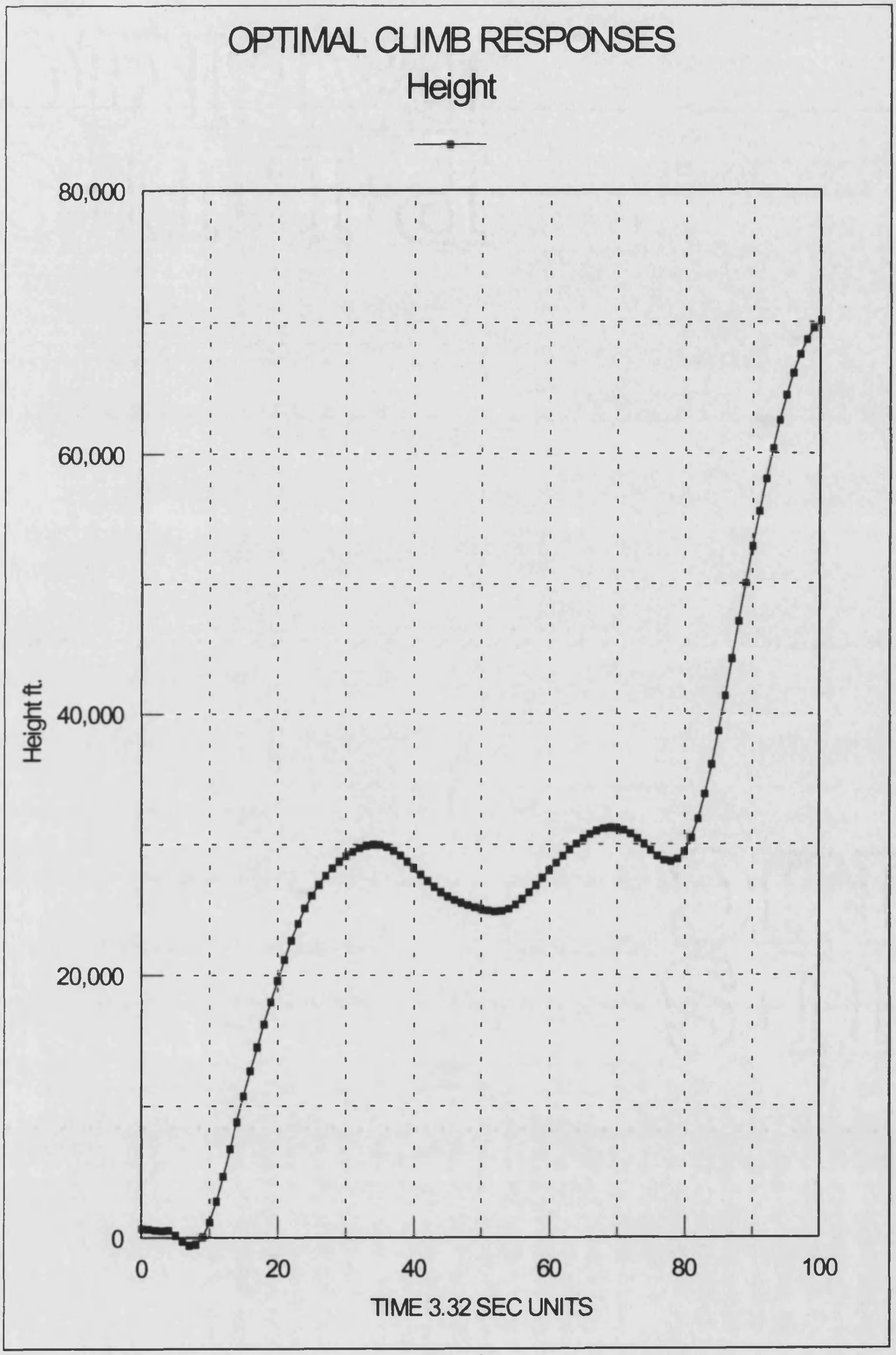

OPTIMAL CLIMB RESPONSES
Velocity
2,000
1,500
1,000
500
0
Velocity ft./sec
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 8



OPTIMAL CLIMB RESPONSES
Flight-Path Angle
Flight-Path Angle Rad.
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 9



OPTIMAL CLIMB RESPONSES
Height
Height ft.
80,000
60,000
40,000
20,000
0
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 10

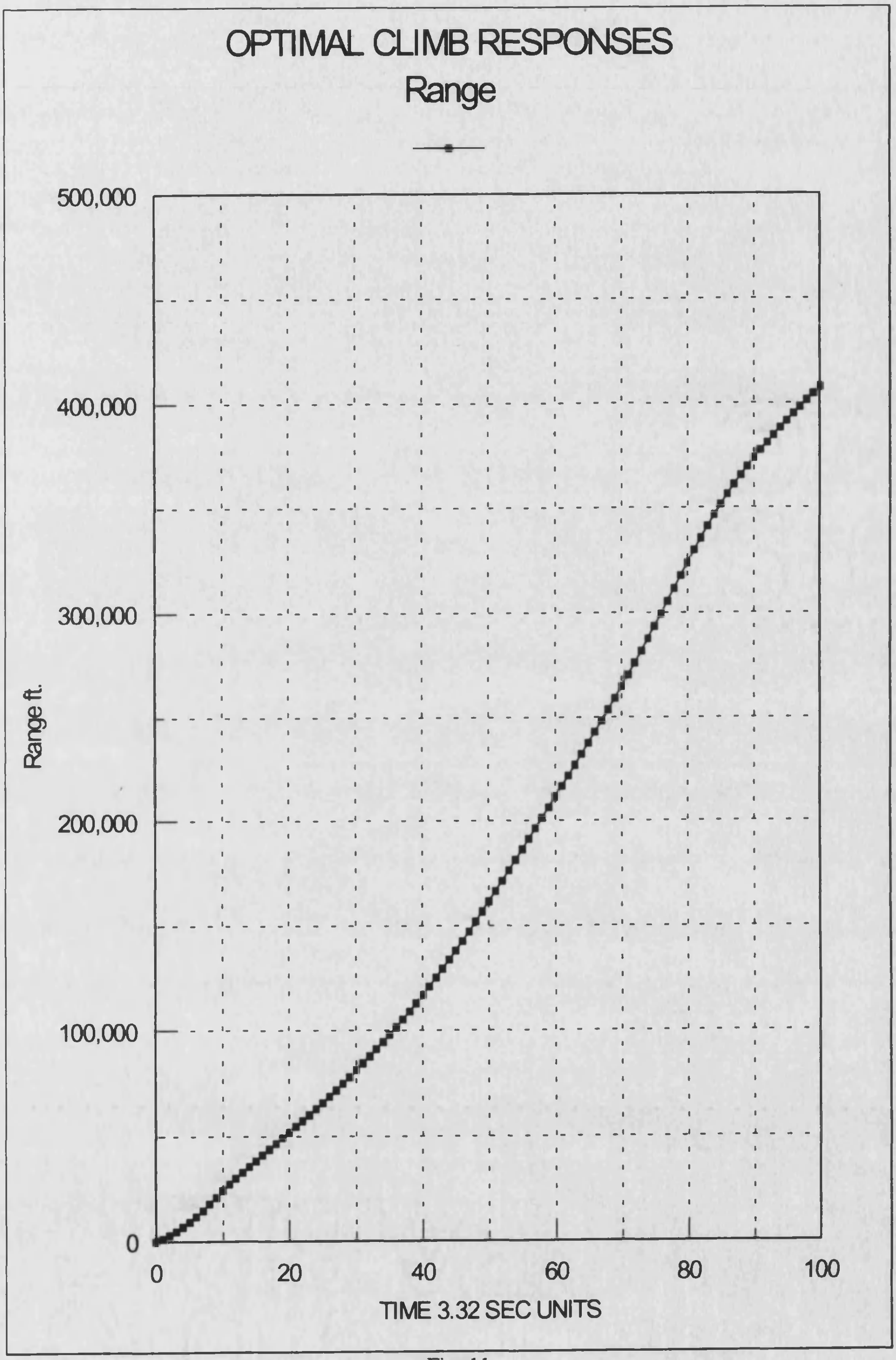
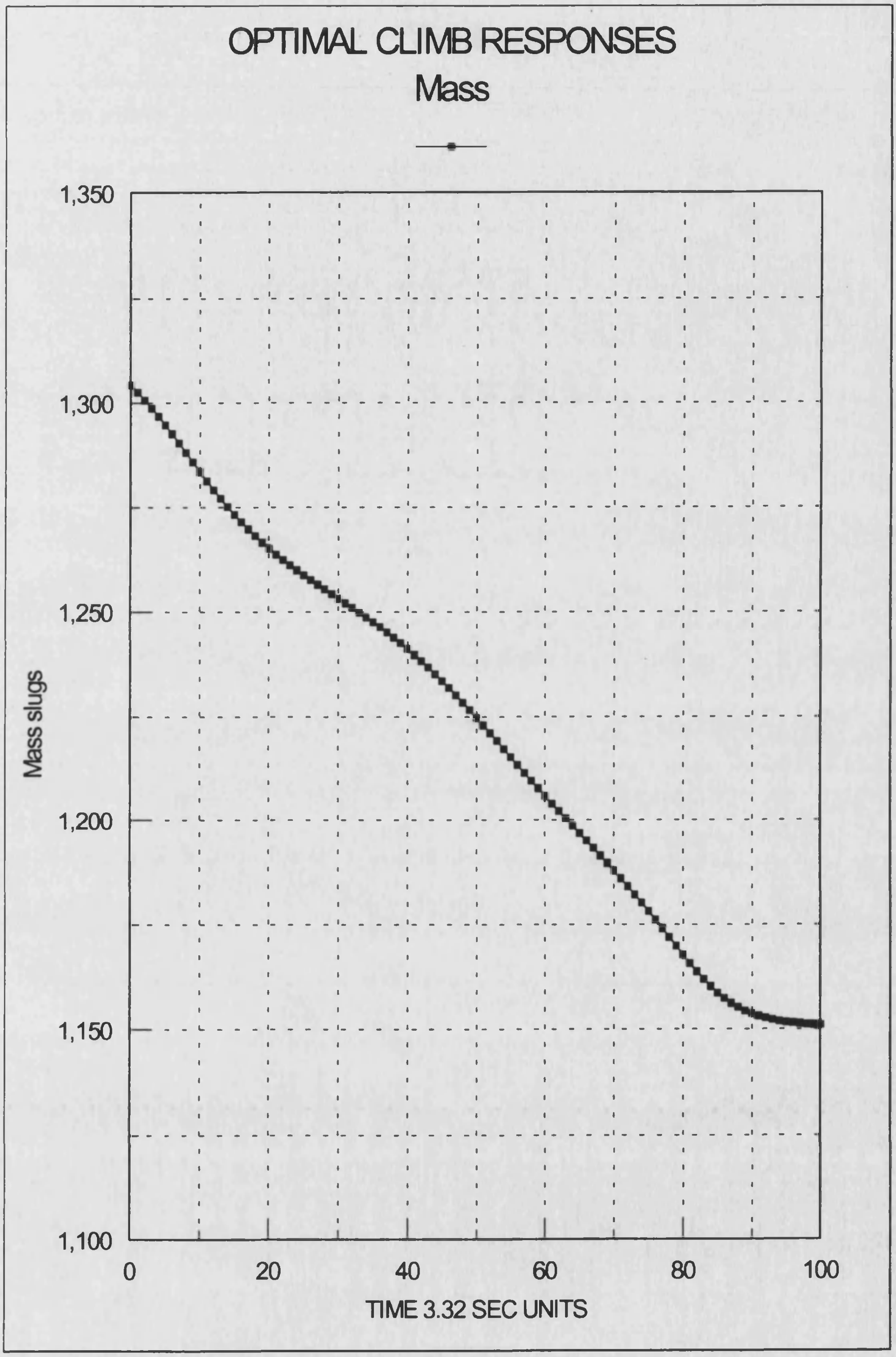

OPTIMAL CLIMB RESPONSES
Range
Range ft.
500,000
400,000
300,000
200,000
100,000
0
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 11



OPTIMAL CLIMB RESPONSES
Mass
1,350
1,300
1,250
1,200
1,150
1,100
Mass slugs
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 12

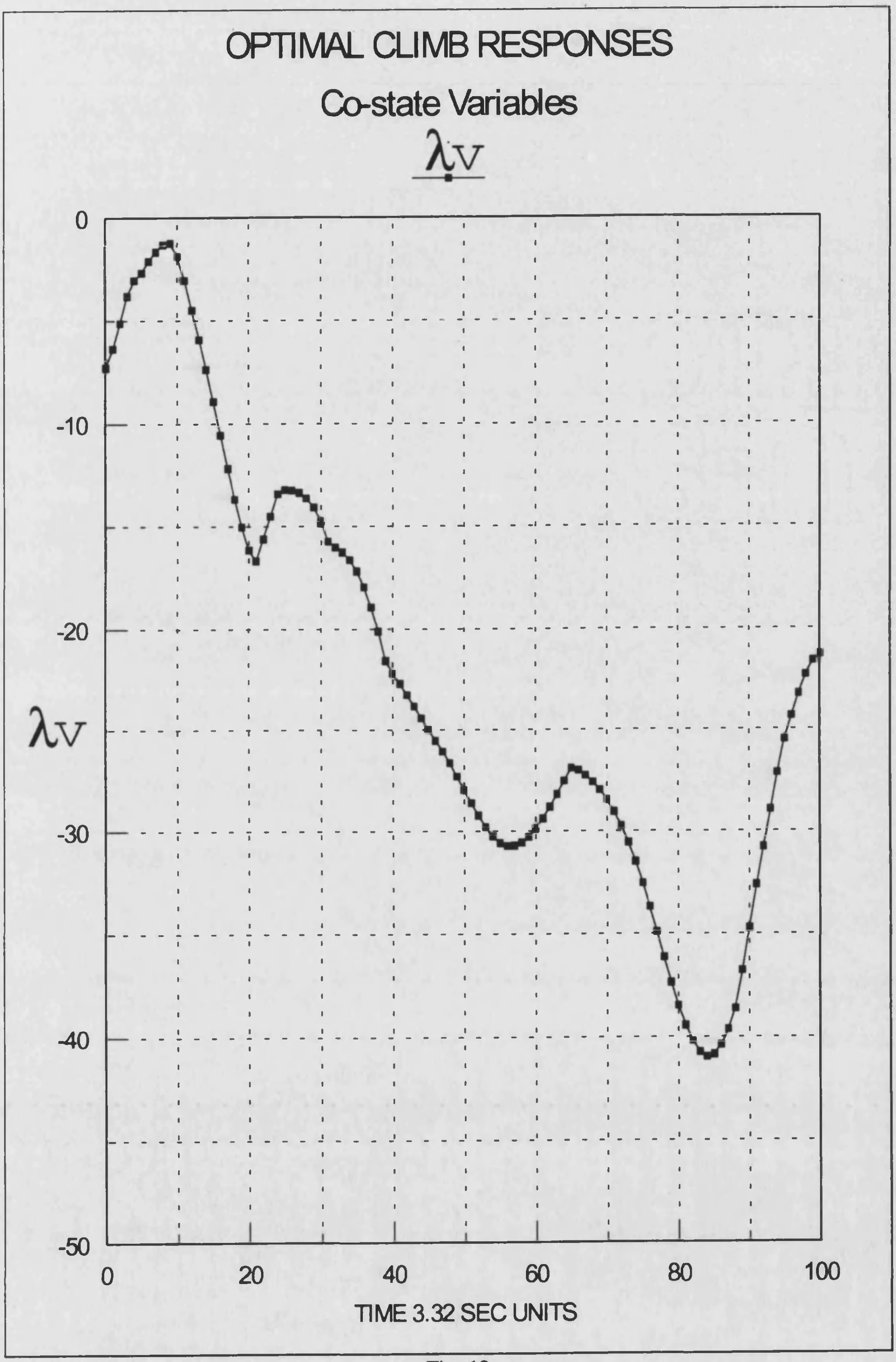

OPTIMAL CLIMB RESPONSES
Co-state Variables
λV
0
-10
-20
λV
-30
-40
-50
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 13

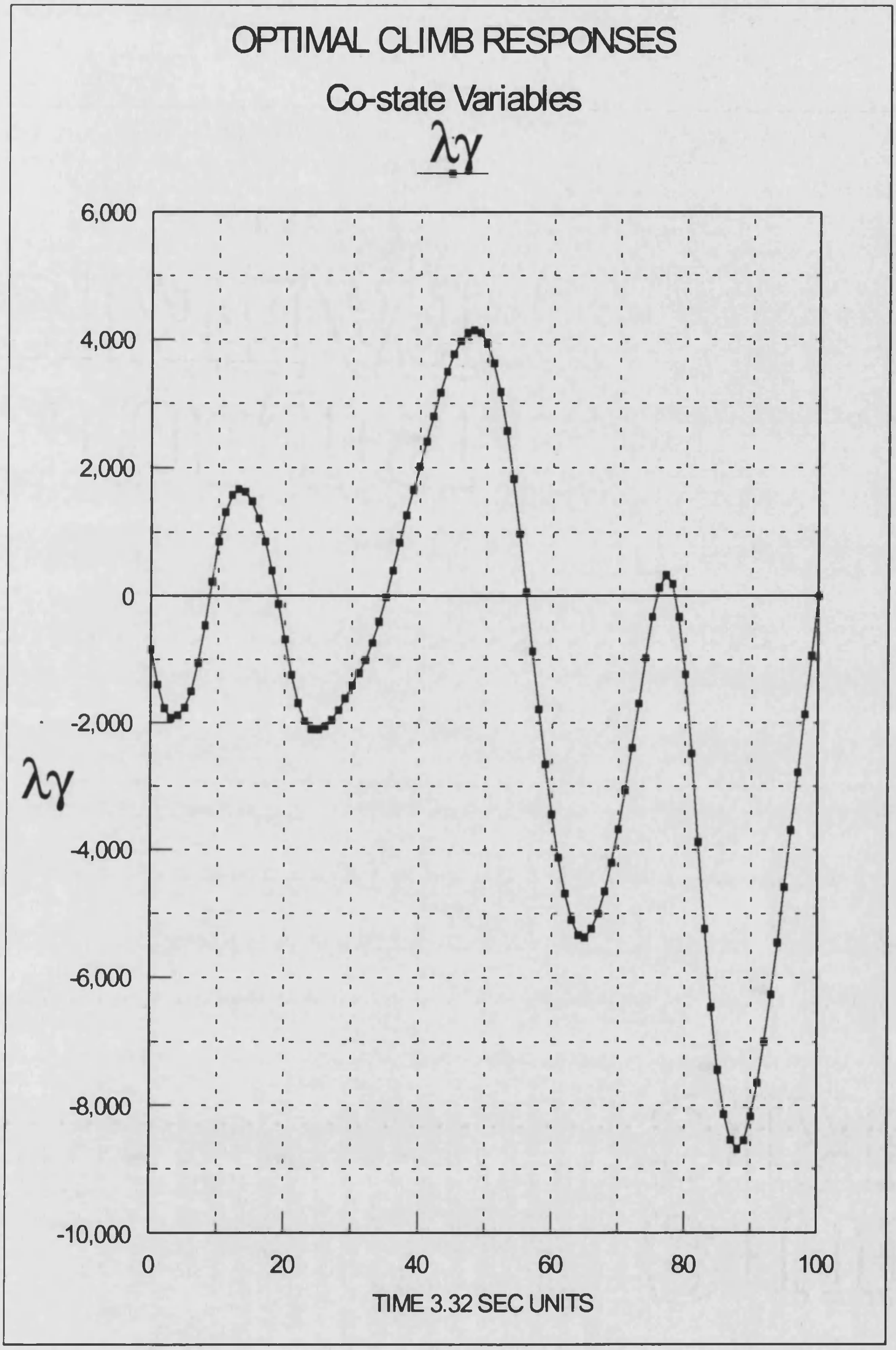

OPTIMAL CLIMB RESPONSES
Co-state Variables
$\lambda\gamma$
6,000
4,000
2,000
0
-2,000
-4,000
-6,000
-8,000
-10,000
$\lambda\gamma$
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 14

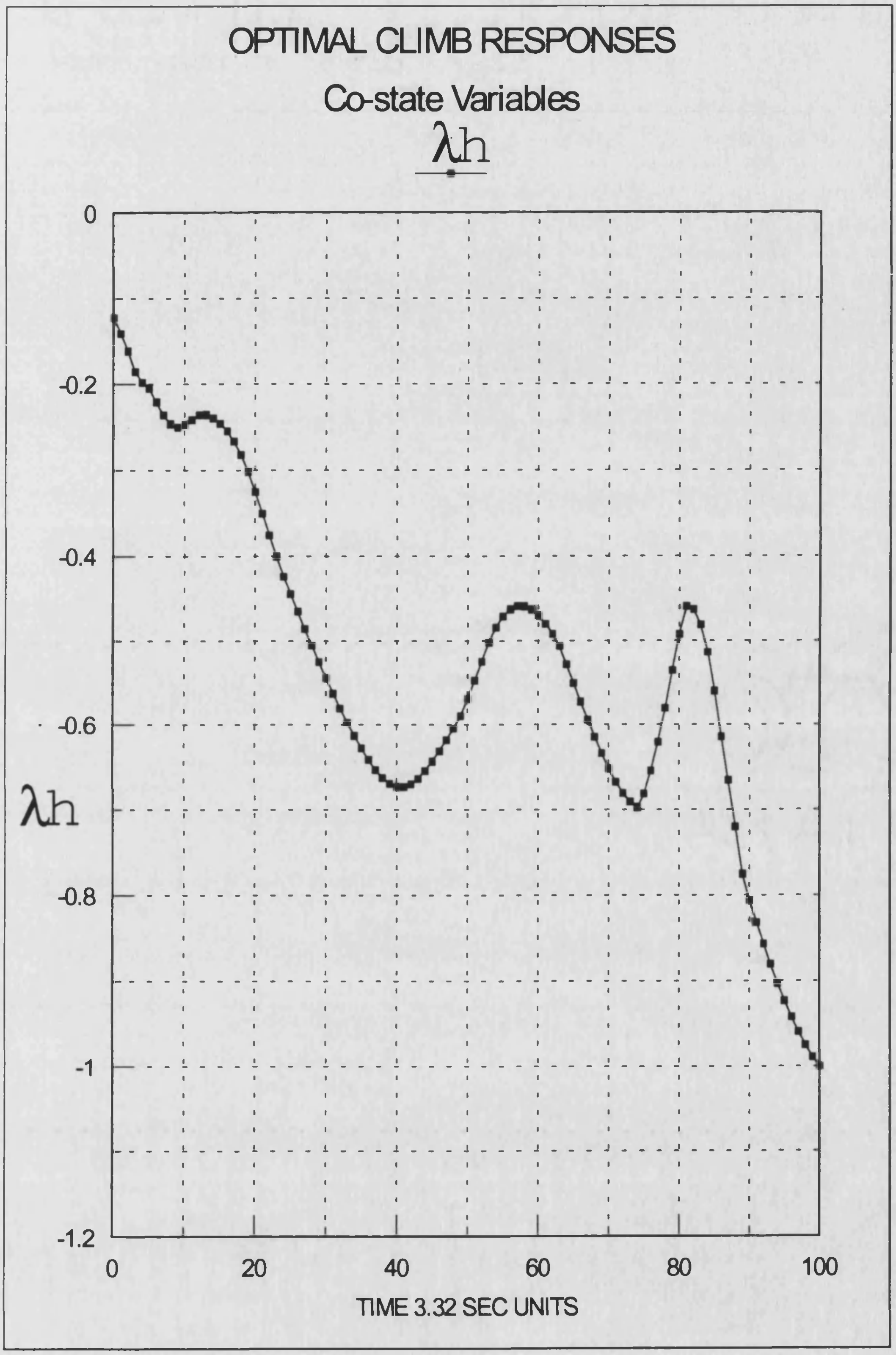
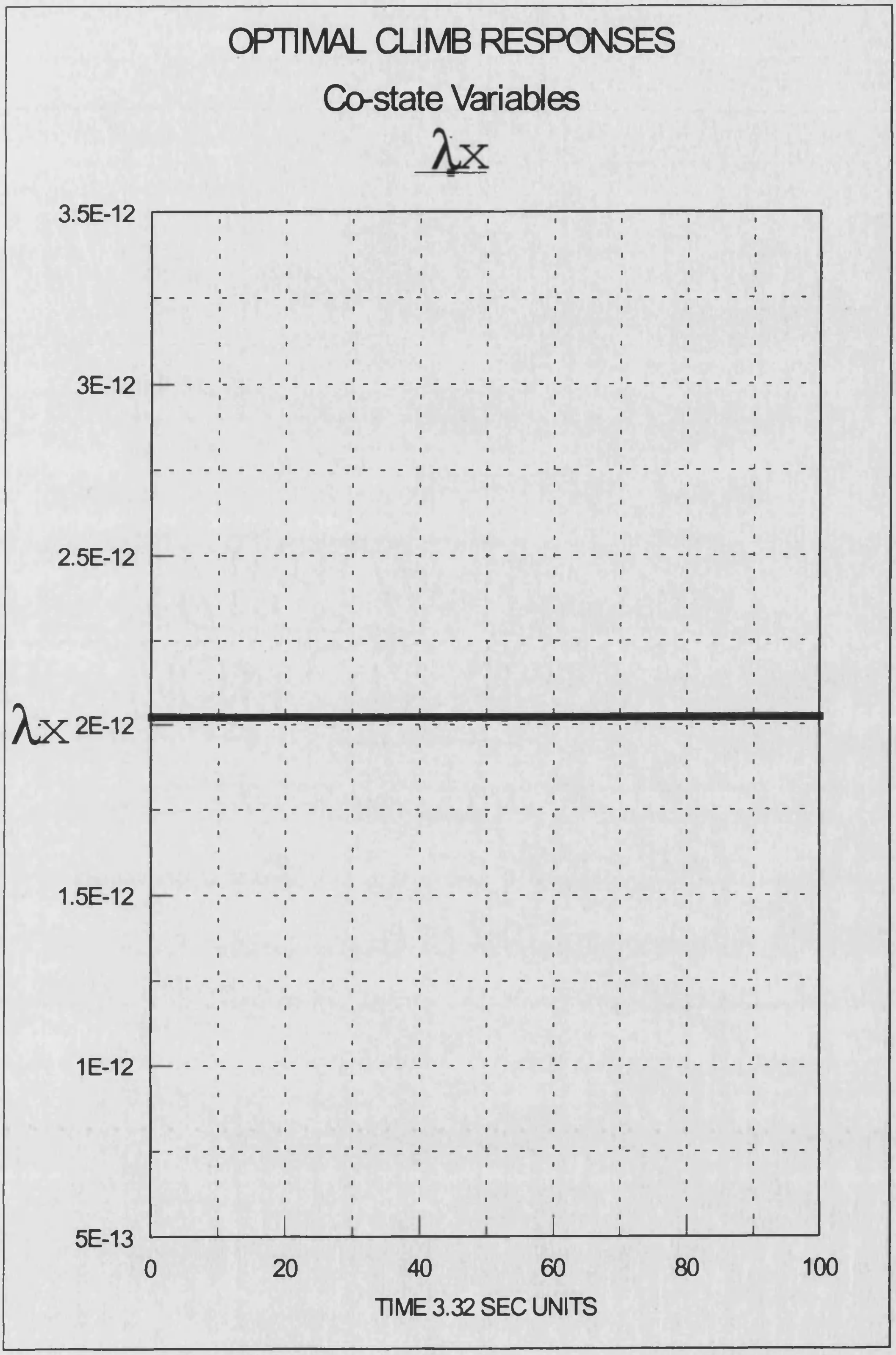

OPTIMAL CLIMB RESPONSES
Co-state Variables
λh
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
λh
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig.15



OPTIMAL CLIMB RESPONSES
Co-state Variables
λx
3.5E-12
3E-12
2.5E-12
λx
2E-12
1.5E-12
1E-12
5E-13
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 16

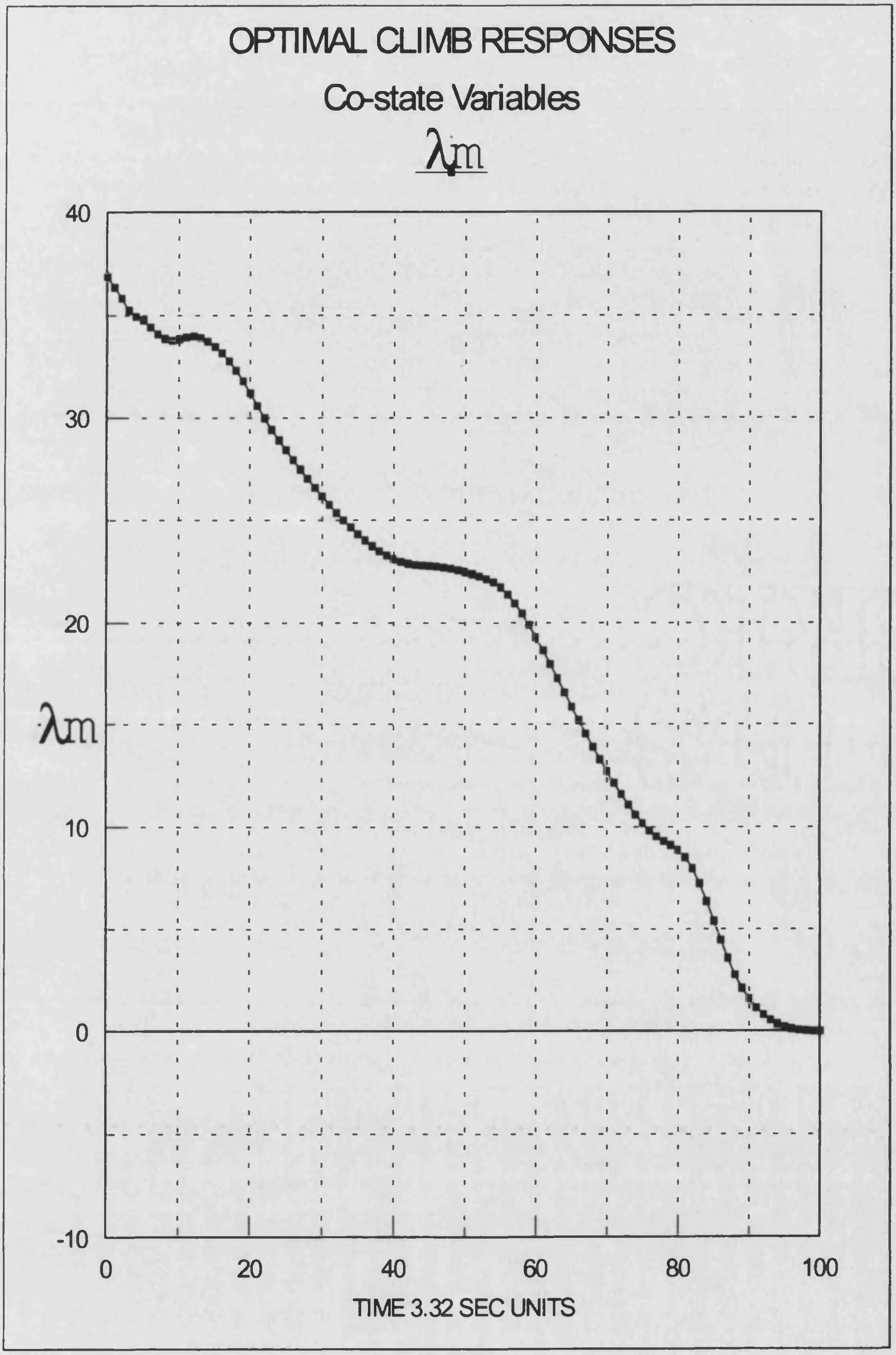

OPTIMAL CLIMB RESPONSES
Co-state Variables
λm
40
30
20
λm
10
0
-10
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 17

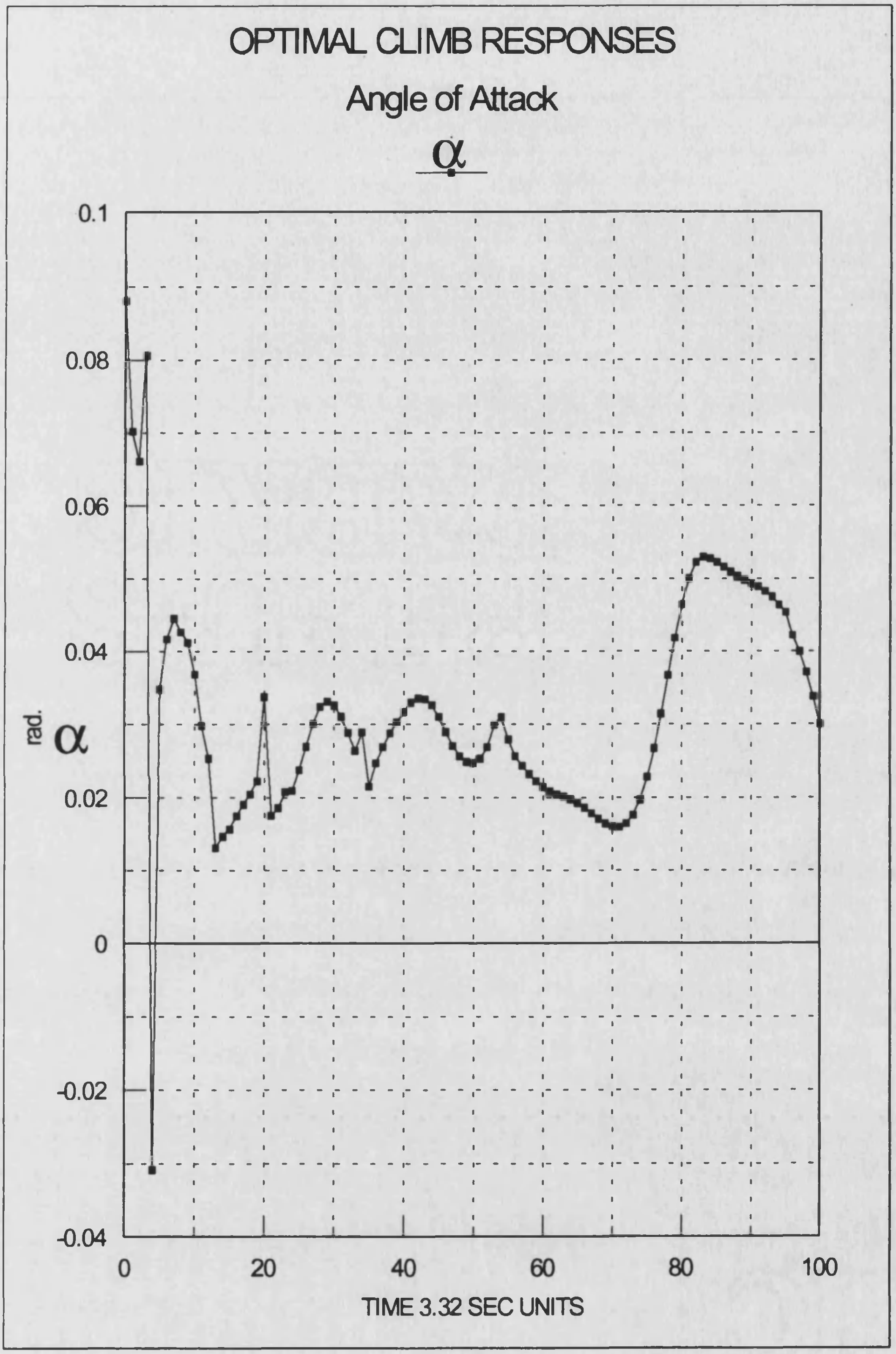

OPTIMAL CLIMB RESPONSES
Angle of Attack
α
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
rad.
α
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 18

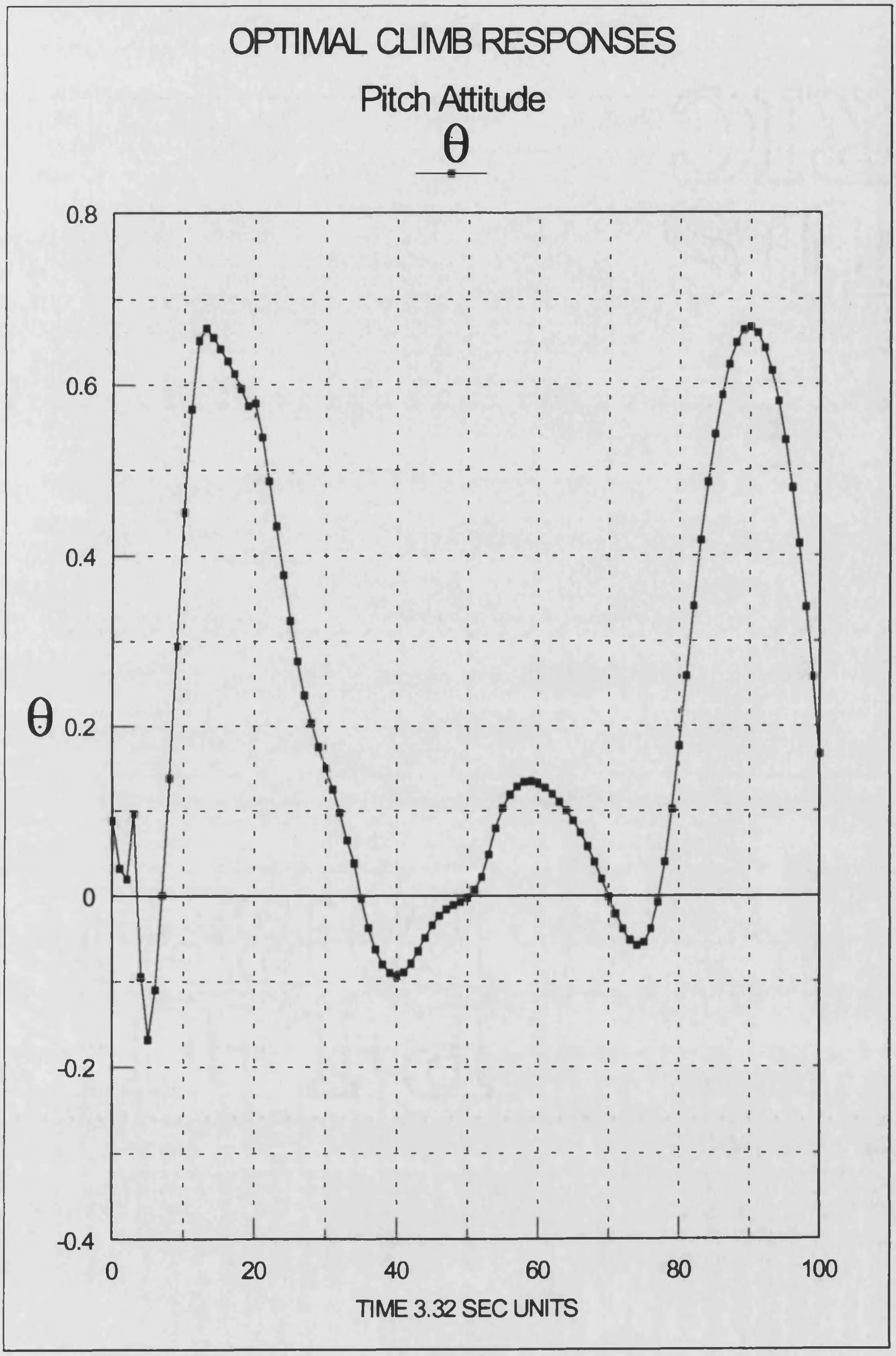

OPTIMAL CLIMB RESPONSES
Pitch Attitude
θ
0.8
0.6
0.4
θ 0.2
0
-0.2
-0.4
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 19

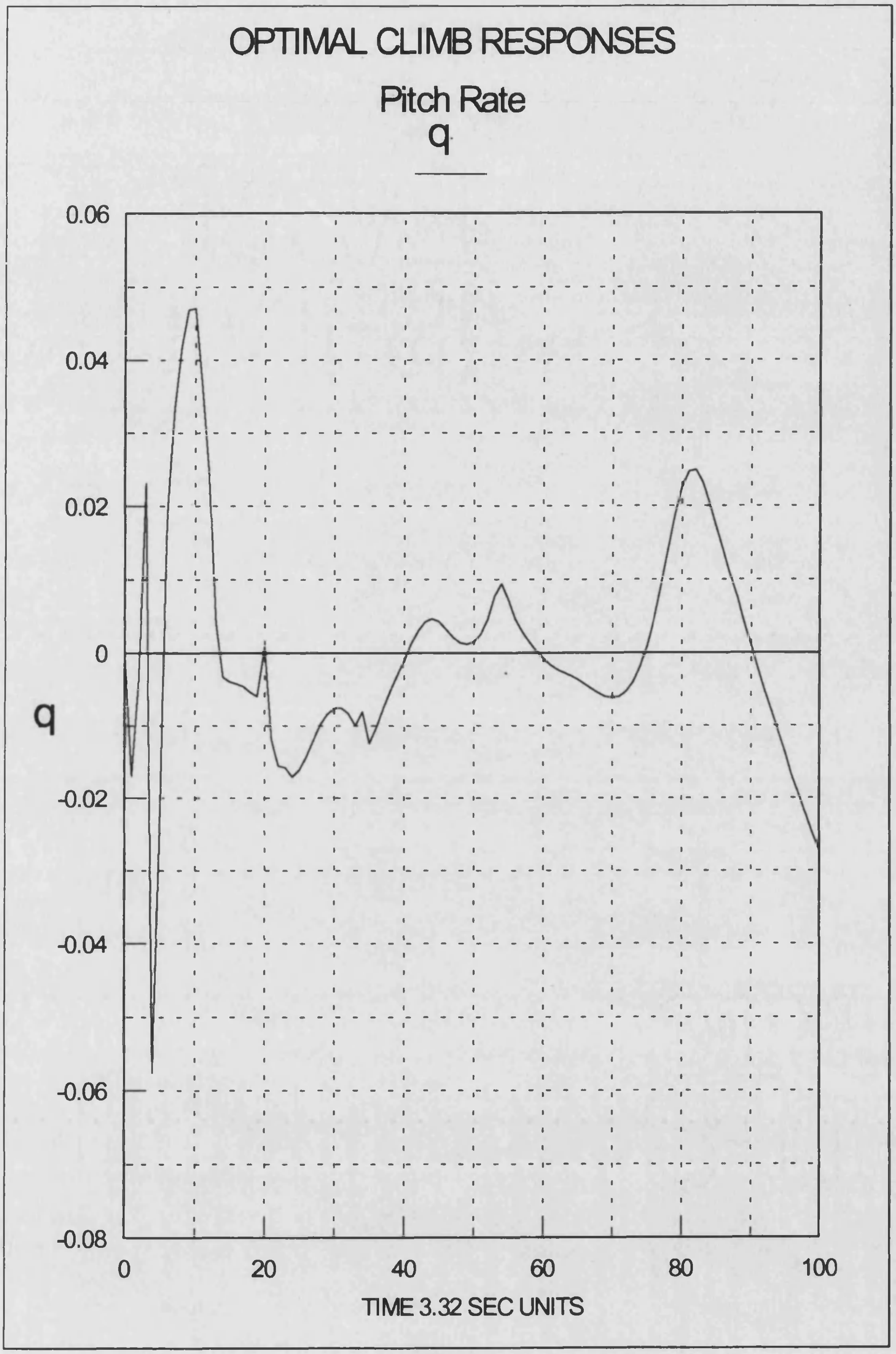
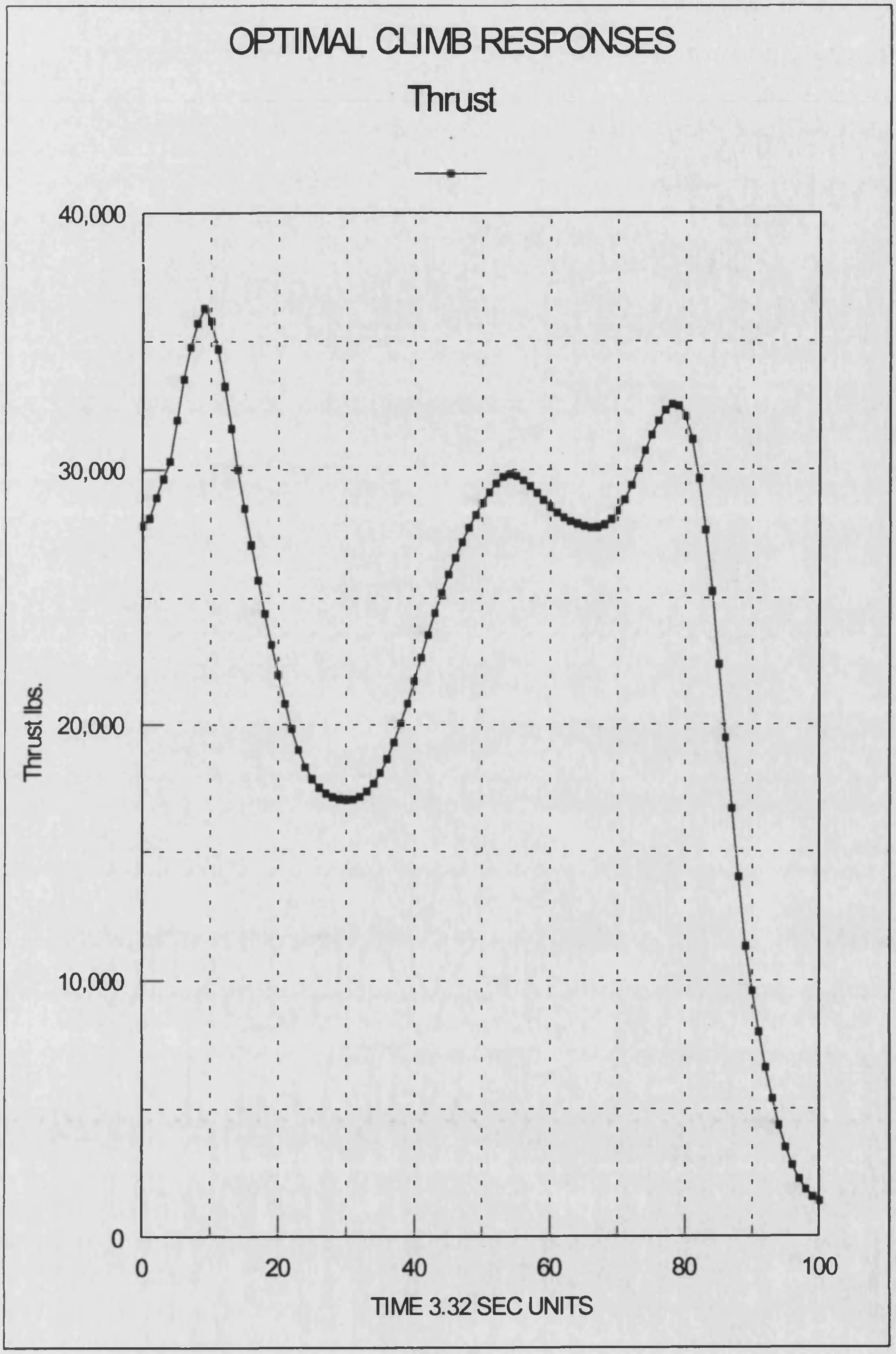

OPTIMAL CLIMB RESPONSES
Pitch Rate
q
0.06
0.04
0.02
0
q
-0.02
-0.04
-0.06
-0.08
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 20



OPTIMAL CLIMB RESPONSES
Thrust
40,000
30,000
20,000
10,000
0
Thrust lbs.
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 21

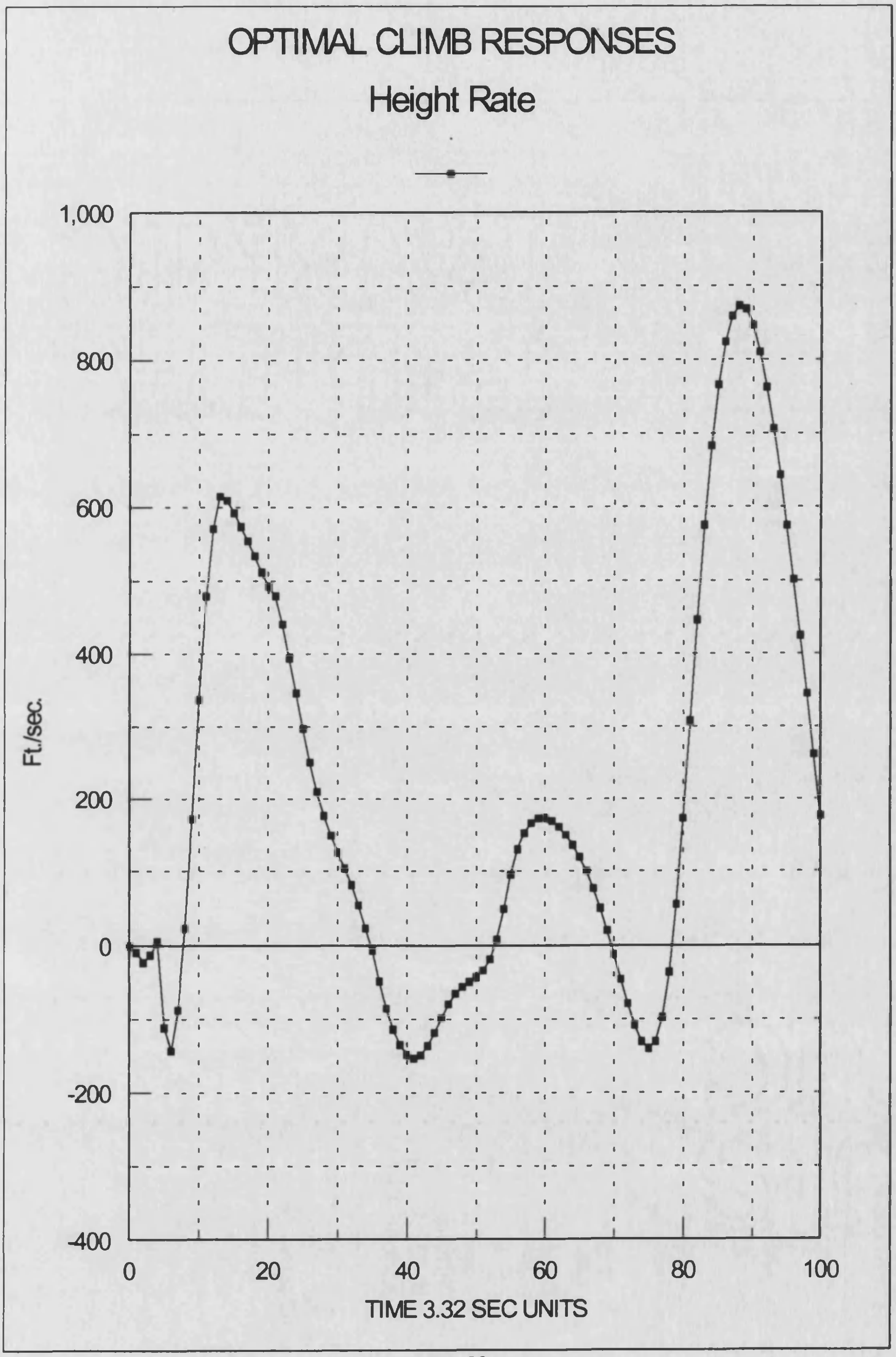
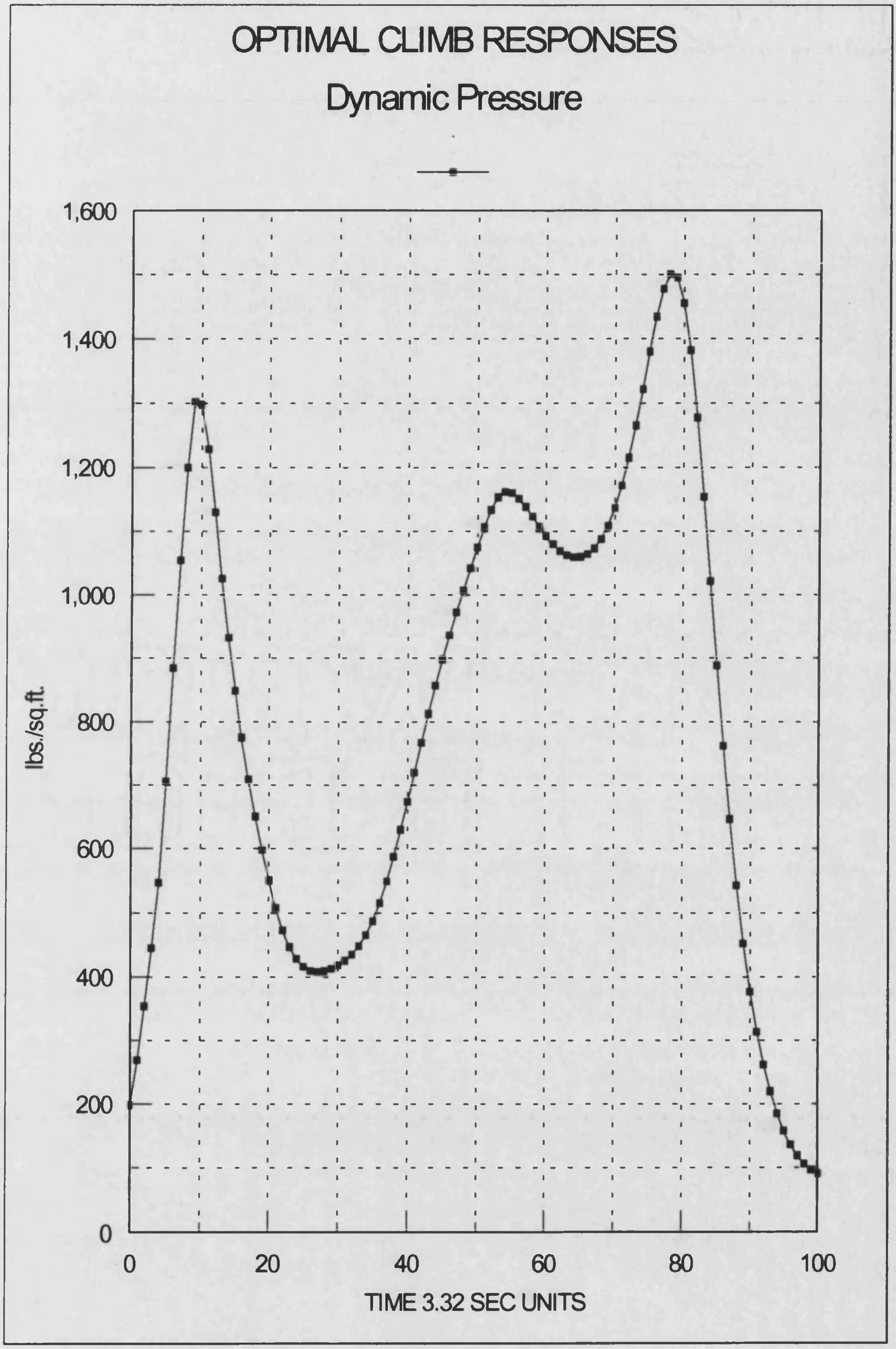

OPTIMAL CLIMB RESPONSES
Height Rate
1,000
800
600
400
200
0
-200
-400
Ft./sec.
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 22



OPTIMAL CLIMB RESPONSES
Dynamic Pressure
1,600
1,400
1,200
1,000
800
600
400
200
0
lbs./sq.ft.
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 23

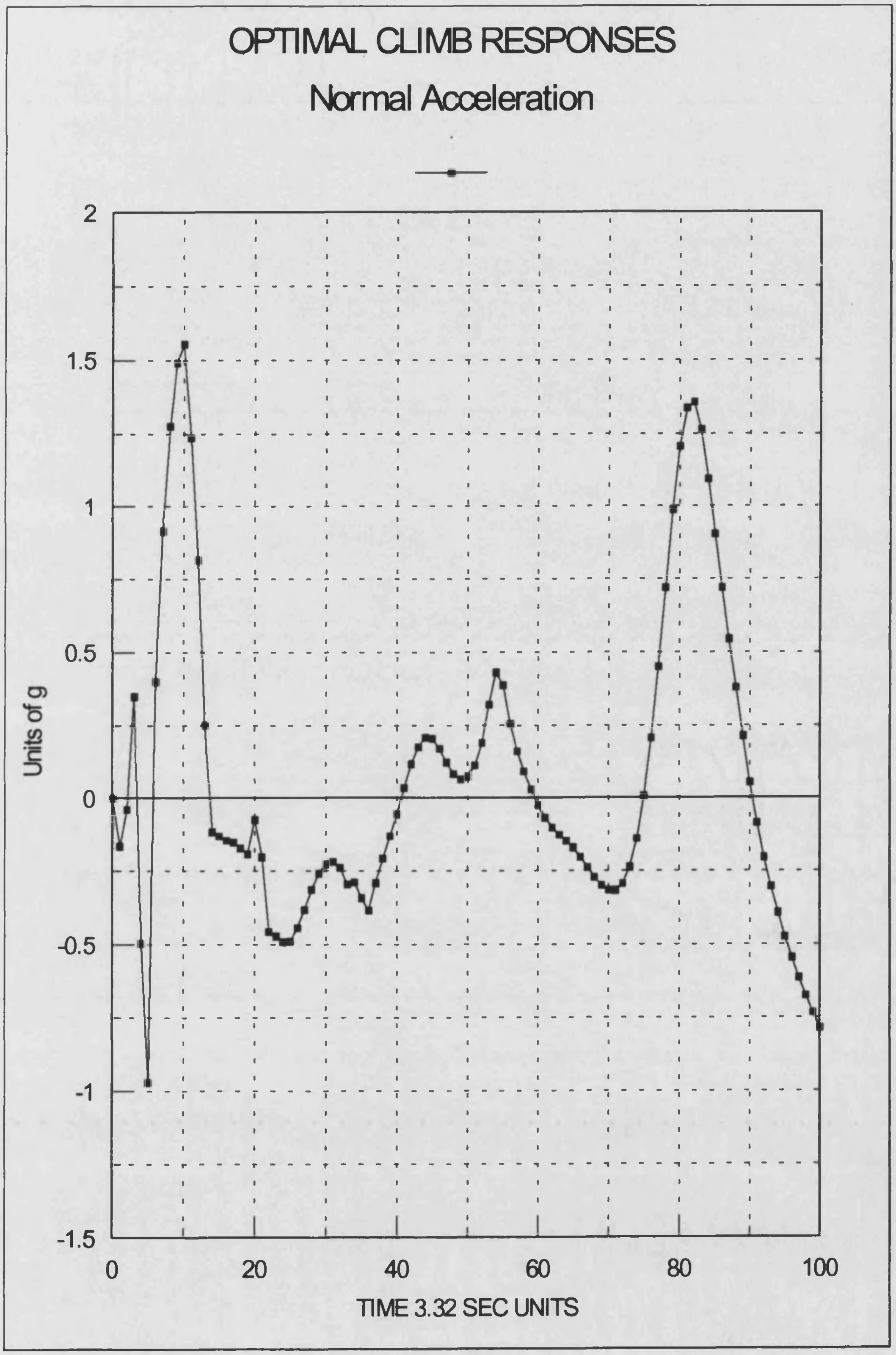

OPTIMAL CLIMB RESPONSES
Normal Acceleration
Units of g
2
1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100
TIME 3.32 SEC UNITS

Fig. 24

Chapter 5

Variation Of Short-Period Pitch Mode Parameters.

Having obtained the solution of the defined optimisation problem by the combined methods of Steepest-Descent and Quasi-linearisation as described in the preceding chapter, the next phase of the investigation was to examine how the aircraft response characteristics varied at different points on the optimal trajectory. For the purpose of this study the aircraft full force equations were linearised about each operating point at every time step in the optimal solution. A second order model representation of the aircraft was assumed in order to define the short-period transfer function of the pitch-rate to elevator response. The phugoid motion was ignored for the purpose of this thesis as in general this is a much longer period than the short period motion. Since the objective of the overall exercise is to maintain the handling qualities of the aircraft as close as possible to an acceptable norm throughout the optimal manoeuvre, and this involves the identification and tracking of the short-period parameters within the transient response time of the system, this simplification of omitting the phugoid mode is justified as there is much more time available to adapt for the phugoid variations should this be necessary. The additional complexity of increasing the number of parameters to be identified in the aircraft representation did not seem to be warranted at this stage of the investigation.

The short-period representation of the aircraft is given by:

$$\frac{q(s)}{\eta(s)} = \frac{(Z_\eta M_w - M_\eta Z_w)\{1 + \dfrac{(M_\eta + Z_\eta M_{\dot{w}})}{(Z_\eta M_w - M_\eta Z_w)} s\}}{s^2 - (Z_w + M_q + M_{\dot{w}} V)s + (Z_w M_q - V M_w)}$$

where the stability derivatives are given as:

$$M_q = \frac{\rho S V \overline{c}^2}{4I_{yy}} C_{mq}; M_w = \frac{\rho S V \overline{c}}{2I_{yy}} C_{m\alpha}; M_{\dot{w}} = \frac{\rho S \overline{c}^2}{4I_{yy}} C_{m\dot{\alpha}}; M_\eta = \frac{\rho S V^2 \overline{c}}{2I_{yy}} C_{m\eta};$$

$$Z_w = -\frac{\rho S V}{2m}(C_{L\alpha} + C_D); Z_\eta = -\frac{\rho S V^2}{2m} C_{L\eta}$$

The short-period representation of the aircraft can be expressed in the standard second-order form more commonly used by control engineers as:

$$\frac{q(s)}{\eta(s)} = \frac{K_0 \omega_n^2 \{1 + Ts\}}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

with

$$K_0 = \frac{(Z_\eta M_w - M_\eta Z_w)}{\omega_n^2}; \quad \omega_n^2 = (Z_w M_q - V M_w);$$

$$T = \frac{(M_\eta + Z_\eta M_{\dot{w}})}{(Z_\eta M_w - M_\eta Z_w)}; \quad \xi = \frac{-(Z_w + M_q + M_{\dot{w}} V)}{2\omega_n}$$

The values used for the aerodynamic coefficients in the simulation of the aircraft system are shown in figs. 2, 25, and 26. At every integration step on the optimum trajectory, the values of the stability derivatives were calculated from the aerodynamic coefficients using an interpolation programme to compute the appropriate values of these coefficients at the intermediate values of mach number pertaining at that point on the optimum trajectory. The aircraft system parameters as defined above were subsequently computed from the stability derivatives. The variation of these system parameters with respect to dynamic pressure and mach number on the optimal climb trajectory is shown in figs. 27 to 30. From these graphs it is readily seen that the aircraft parameters are highly non-linear with respect to dynamic pressure and mach number and indeed are not single valued with respect to these variables. This implies that it is difficult to obtain a single valued gain schedule with which to modify any command stability augmentation controller which may be used in an attempt to obtain a uniform closed-loop pitch rate per elevator response throughout the optimal trajectory.

The variation in the values of the aircraft system parameters as the aircraft flies the optimal trajectory are shown in figs 27 to 30. It is readily seen that the undamped natural frequency varies through a range of approximately six to one . The damping ratio varies from approximately 0.4 at the beginning of the manoeuvre to 0.06 towards the end. The lead time constant varies by a factor of five while the steady state gain varies from in excess of ten to one.

The effect of these parameter variations on the short period open-loop pitch rate per elevator transient step response of the basic aircraft during the optimal trajectory are shown in figures 31 to 51 every 16.6 seconds on the optimal climb trajectory.

It is obvious that if a uniform response is to be achieved in the short period handling characteristics of the aircraft, a stability augmentation system is required which is capable of adapting to the aircraft system parameter variations throughout the optimum climb trajectory. It has been shown that the variation of the aircraft system parameters with respect to the traditional gain scheduling variables is both non-linear and not single valued and hence any scheduling of the controller to compensate for these parameter variations will be open-loop. At best it will be a linearised approximation to the required controller parameter values to achieve a uniform closed-loop pitch rate response. As this method is open-loop in the adaptation process, the alternative procedure of identifying the aircraft system parameters on-line, and performing a closed-loop adaptive algorithm has been investigated. This is the subject of chapter7.

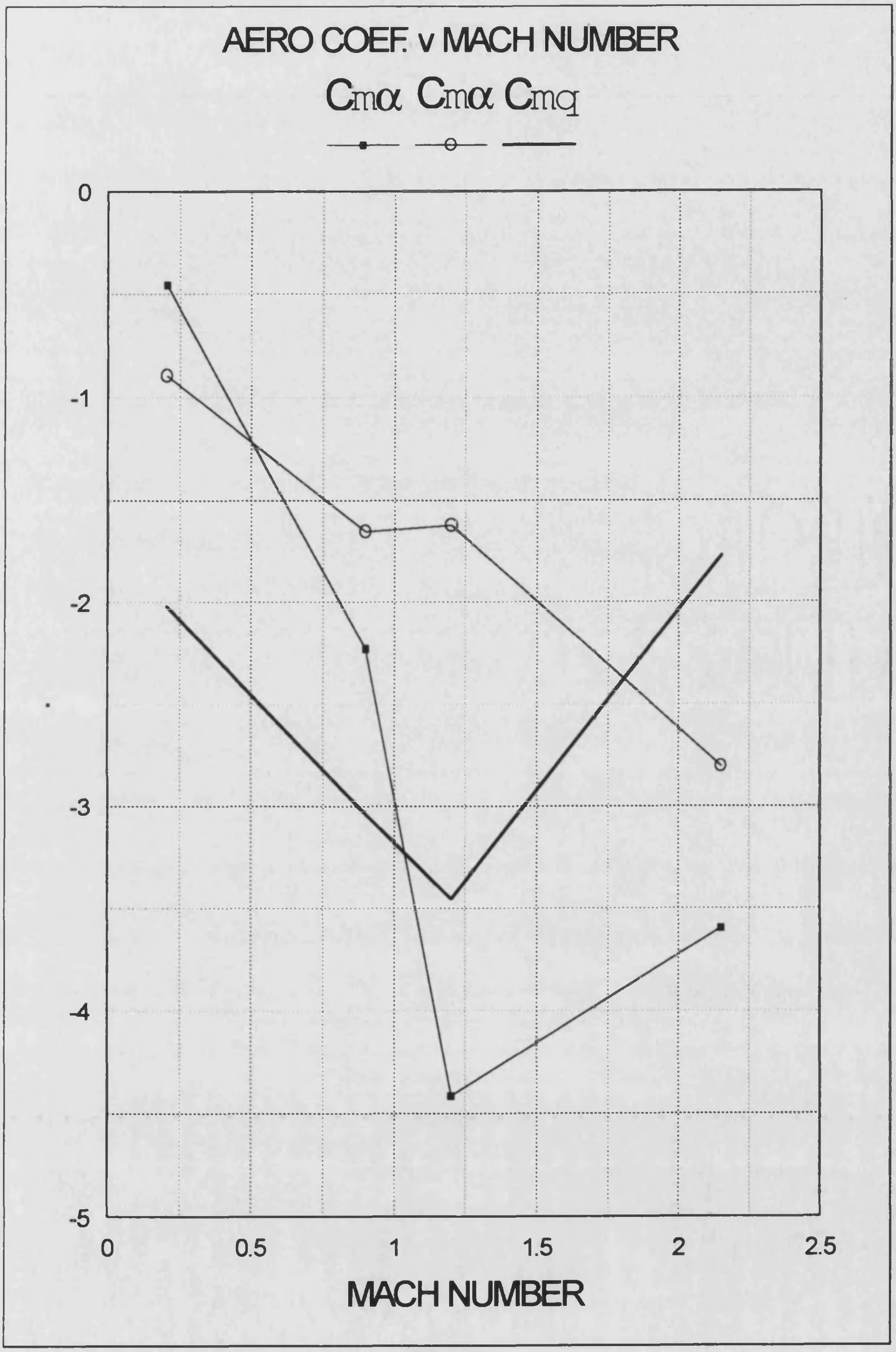
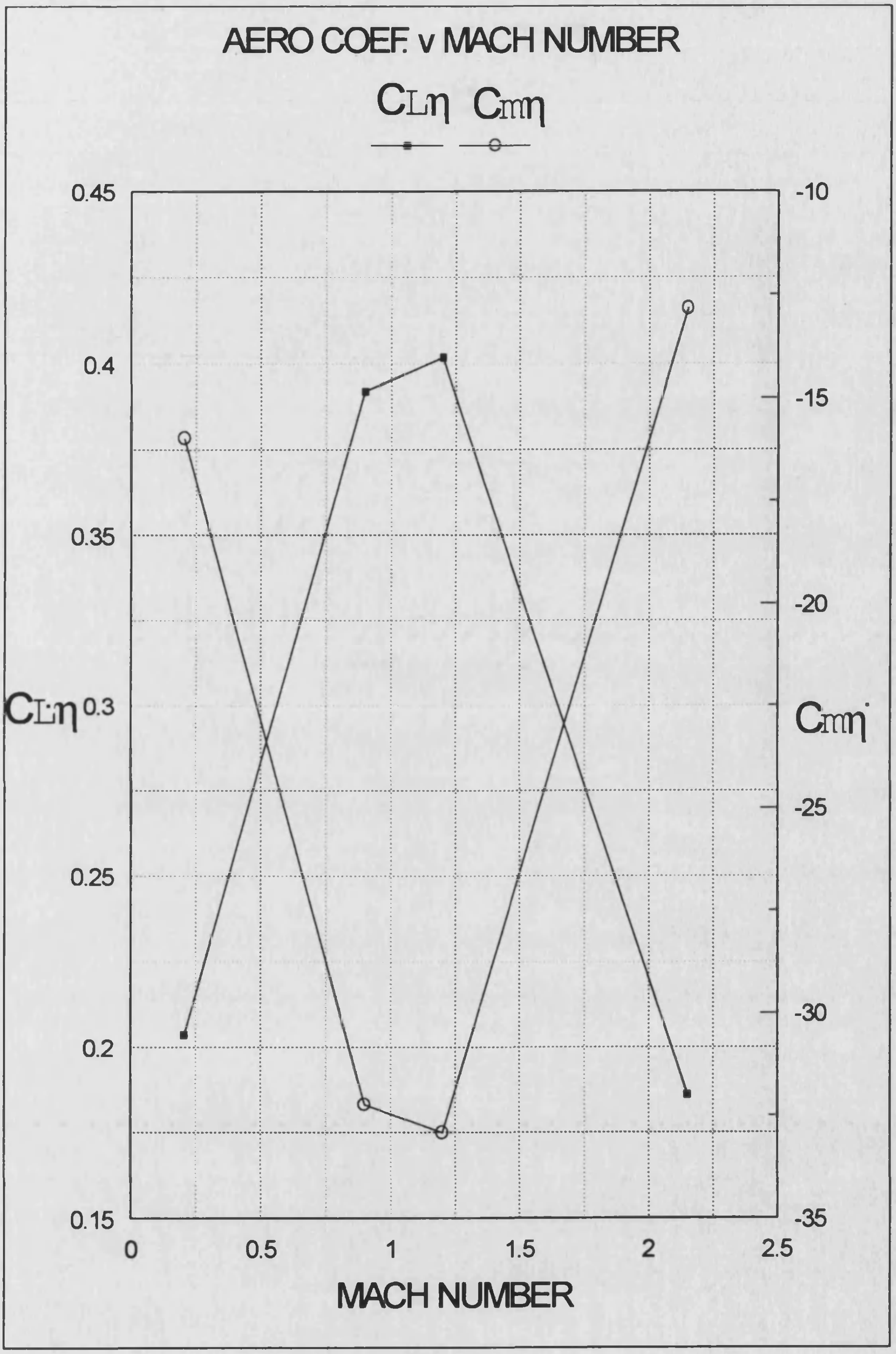
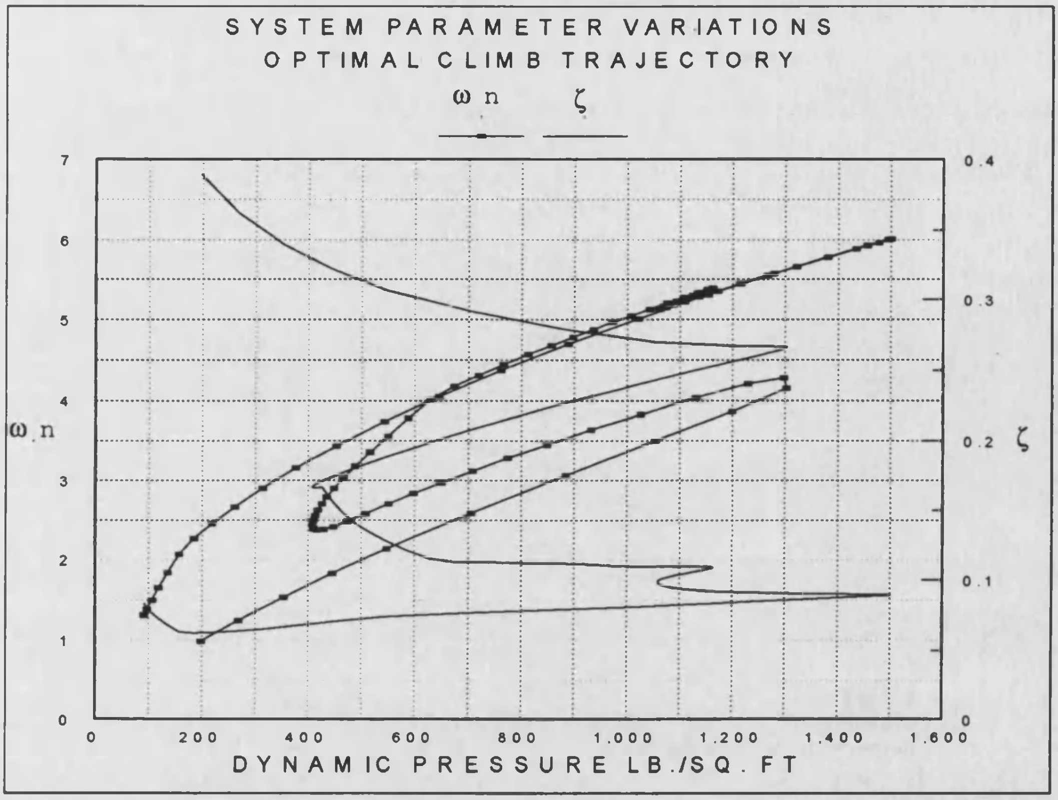

AERO COEF. v MACH NUMBER
Cmα Cmα Cmq
0
-1
-2
-3
-4
-5
0
0.5
1
1.5
2
2.5
MACH NUMBER

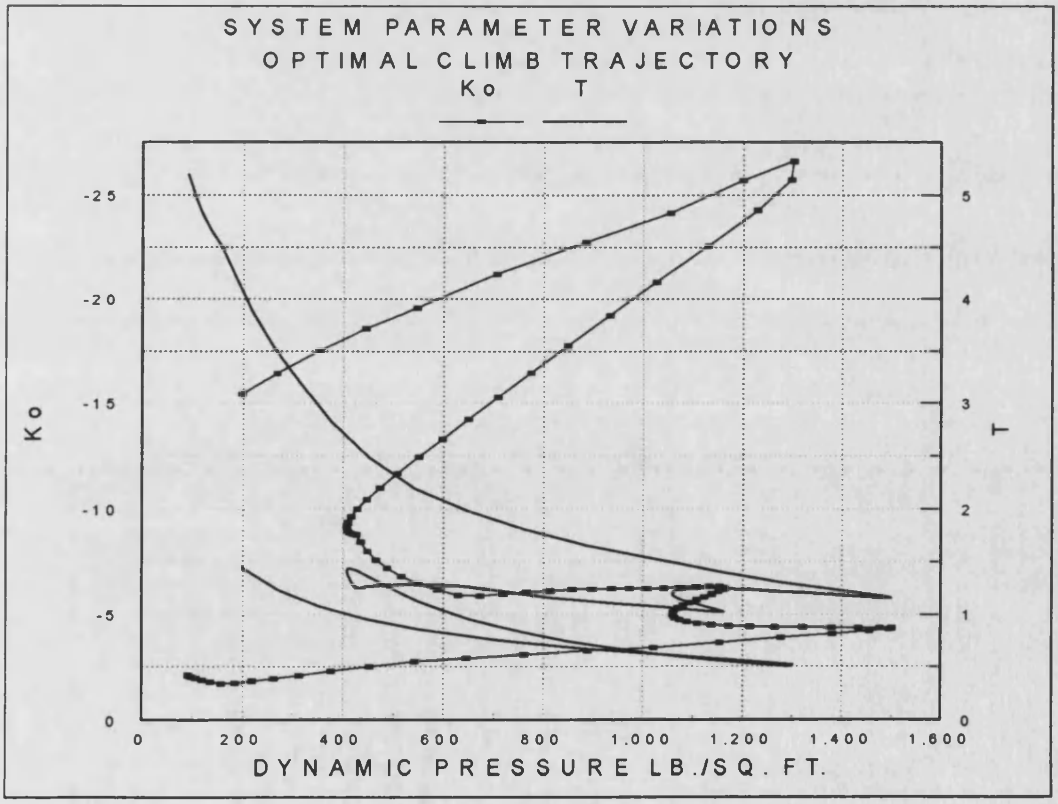
Fig. 25

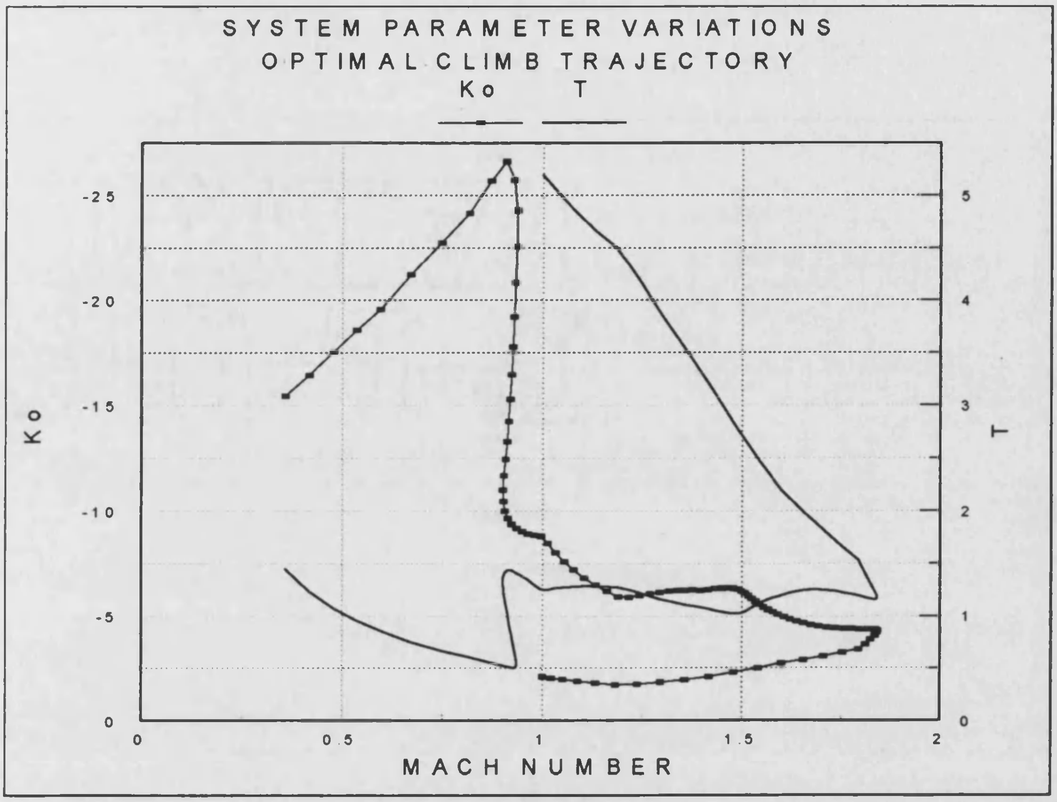


AERO COEF. v MACH NUMBER
CLη Cmη
0.45
0.4
0.35
CLη 0.3
0.25
0.2
0.15
-10
-15
-20
Cmη
-25
-30
-35
0
0.5
1
1.5
2
2.5
MACH NUMBER

Fig. 26

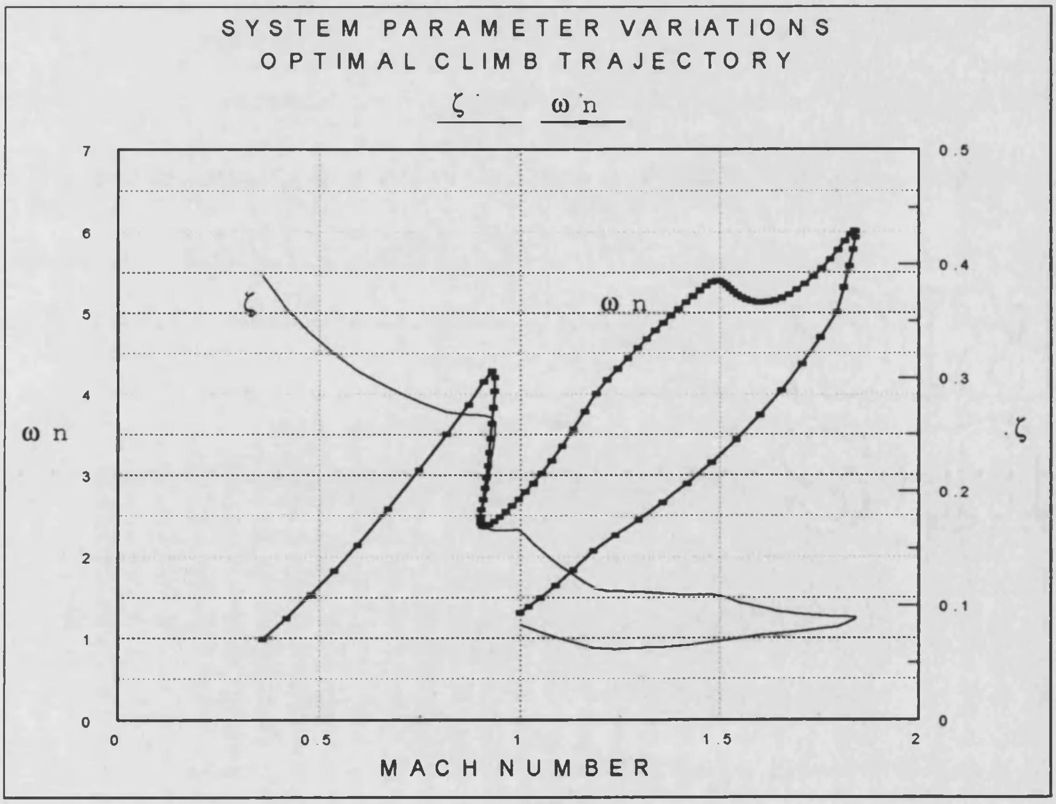

SYSTEM PARAMETER VARIATIONS
OPTIMAL CLIMB TRAJECTORY
ω n
ζ
ω.n
ζ
0
1
2
3
4
5
6
7
0
0.1
0.2
0.3
0.4
0
200
400
600
800
1.000
1.200
1.400
1.600
DYNAMIC PRESSURE LB./SQ. FT
Fig. 27


SYSTEM PARAMETER VARIATIONS
OPTIMAL CLIMB TRAJECTORY
Ko
T
Ko
T
0
-5
-10
-15
-20
-25
0
1
2
3
4
5
0
200
400
600
800
1.000
1.200
1.400
1.600
DYNAMIC PRESSURE LB./SQ. FT.
Fig. 28



SYSTEM PARAMETER VARIATIONS
OPTIMAL CLIMB TRAJECTORY
Ko
T
Ko
T
-25
-20
-15
-10
-5
0
5
4
3
2
1
0
0
0.5
1
1.5
2
MACH NUMBER

Fig. 29



SYSTEM PARAMETER VARIATIONS
OPTIMAL CLIMB TRAJECTORY
ζ
ω n
ζ
ω n
ω n
ζ
7
6
5
4
3
2
1
0
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
MACH NUMBER

Fig. 30

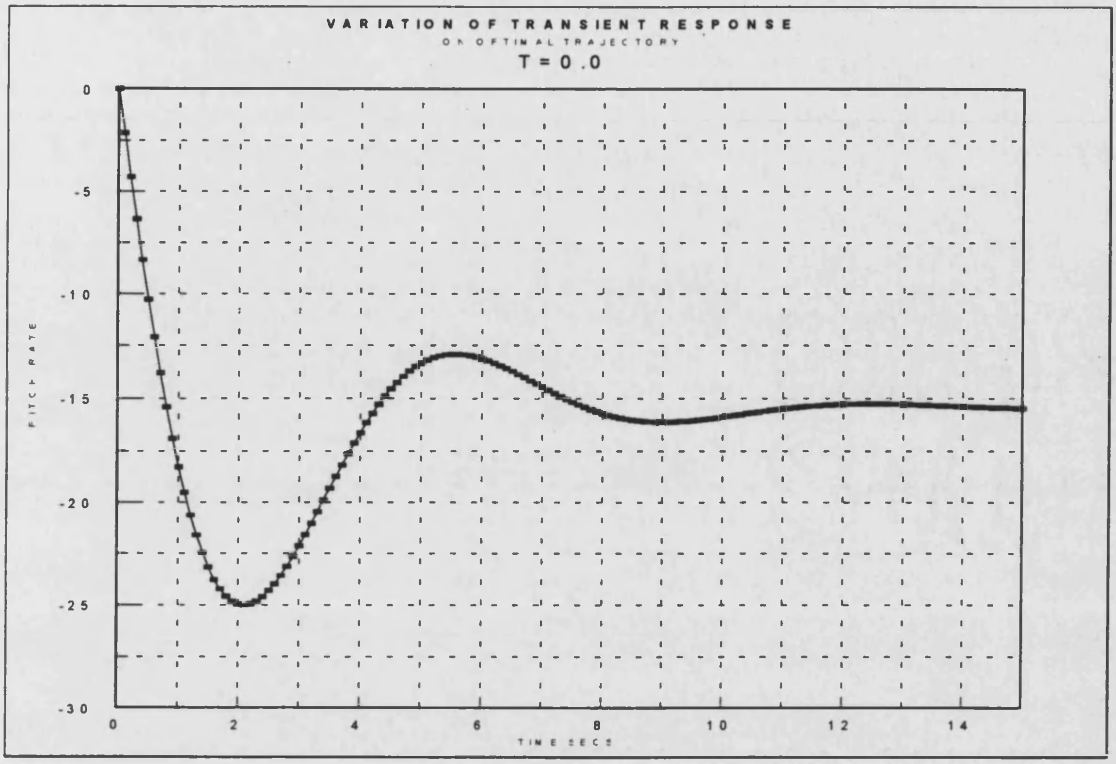

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T = 0.0
0
-5
-10
-15
-20
-25
-30
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 31

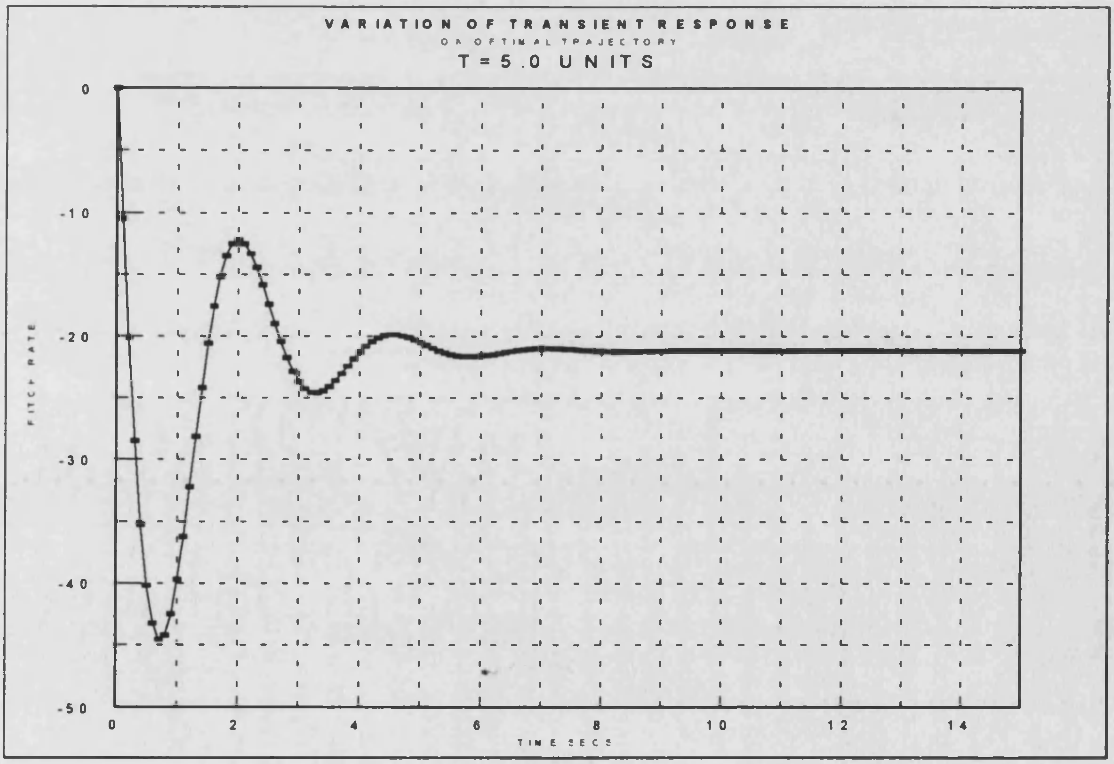

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T = 5.0 UNITS
0
-10
-20
-30
-40
-50
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 32

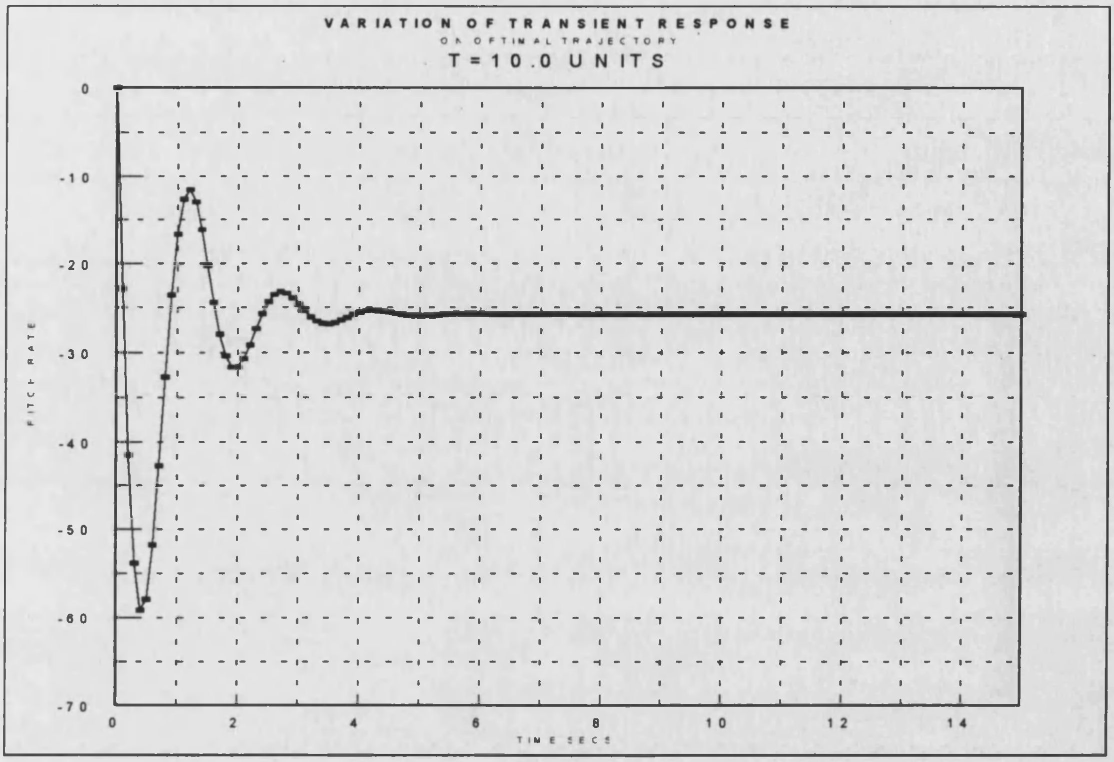

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T=10.0 UNITS
0
-10
-20
-30
-40
-50
-60
-70
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 33

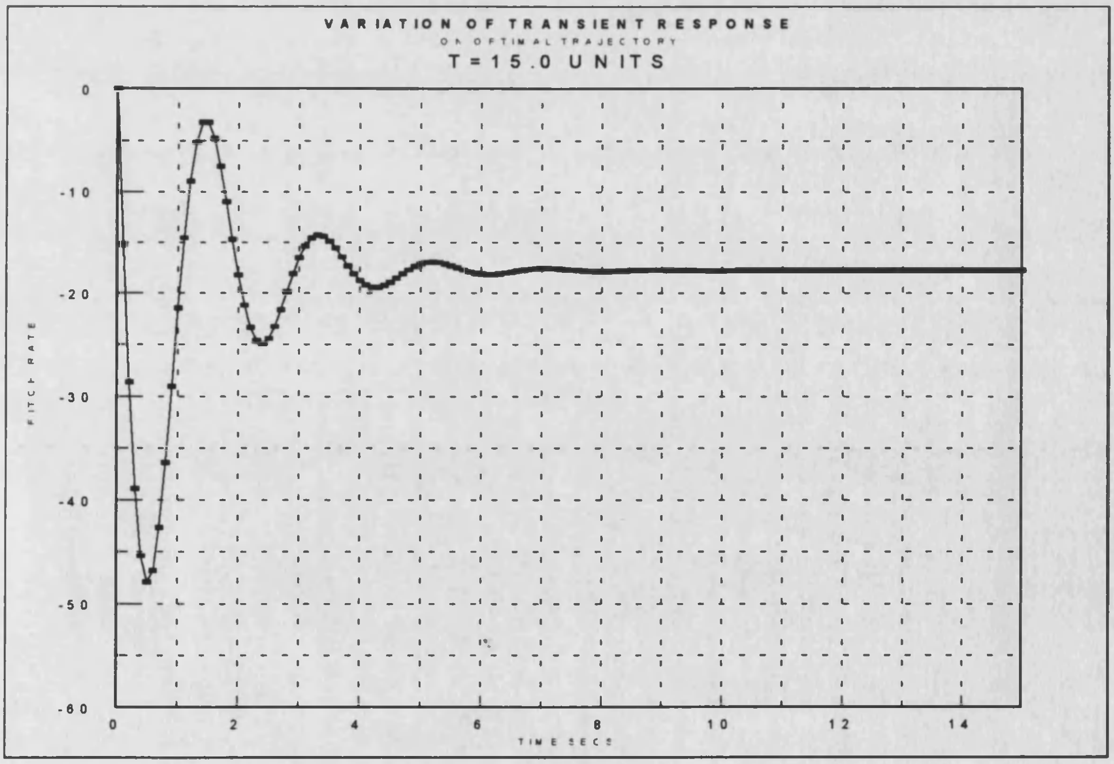

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T=15.0 UNITS
0
-10
-20
-30
-40
-50
-60
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 34

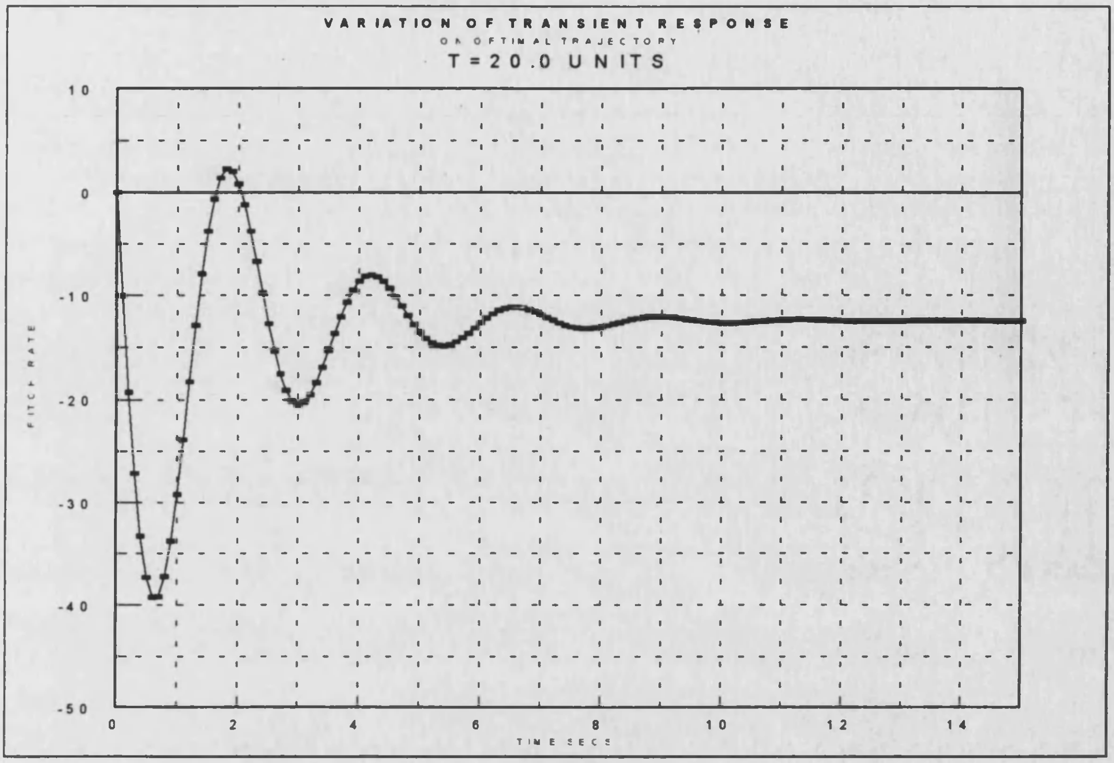

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T = 20.0 UNITS
10
0
-10
-20
-30
-40
-50
0
2
4
6
8
10
12
14
TIME SECS

Fig. 35

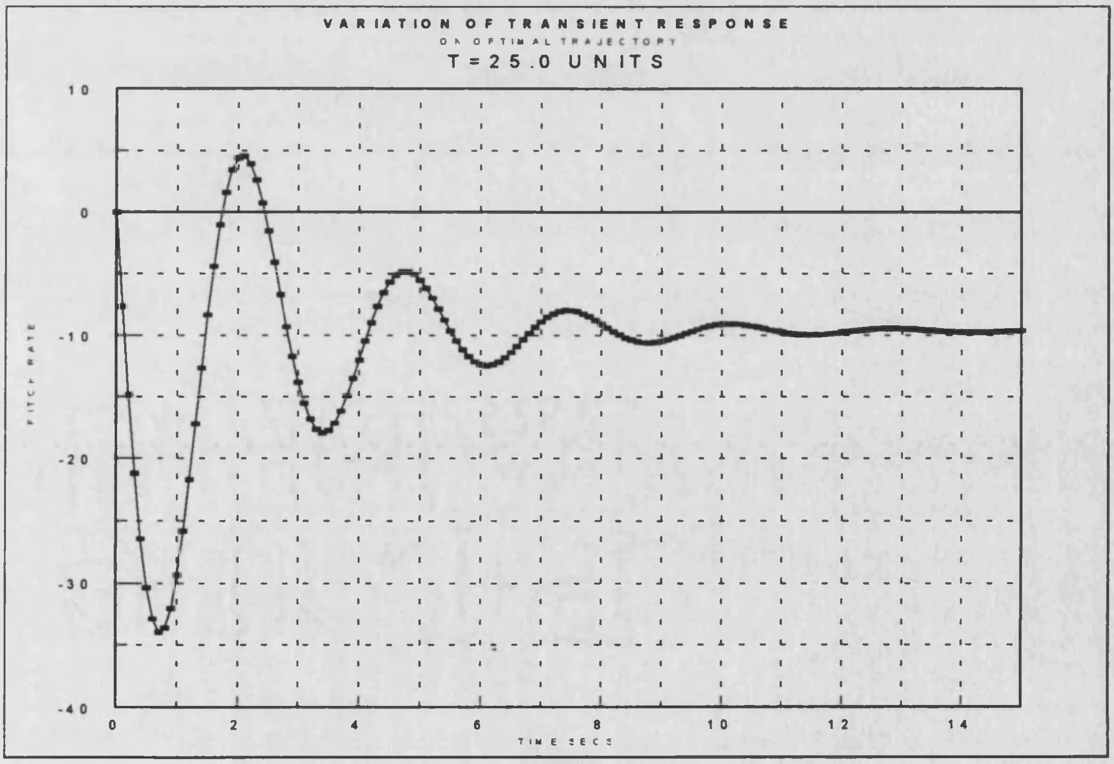

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T = 25.0 UNITS
10
0
-10
-20
-30
-40
0
2
4
6
8
10
12
14
TIME SECS

Fig. 36

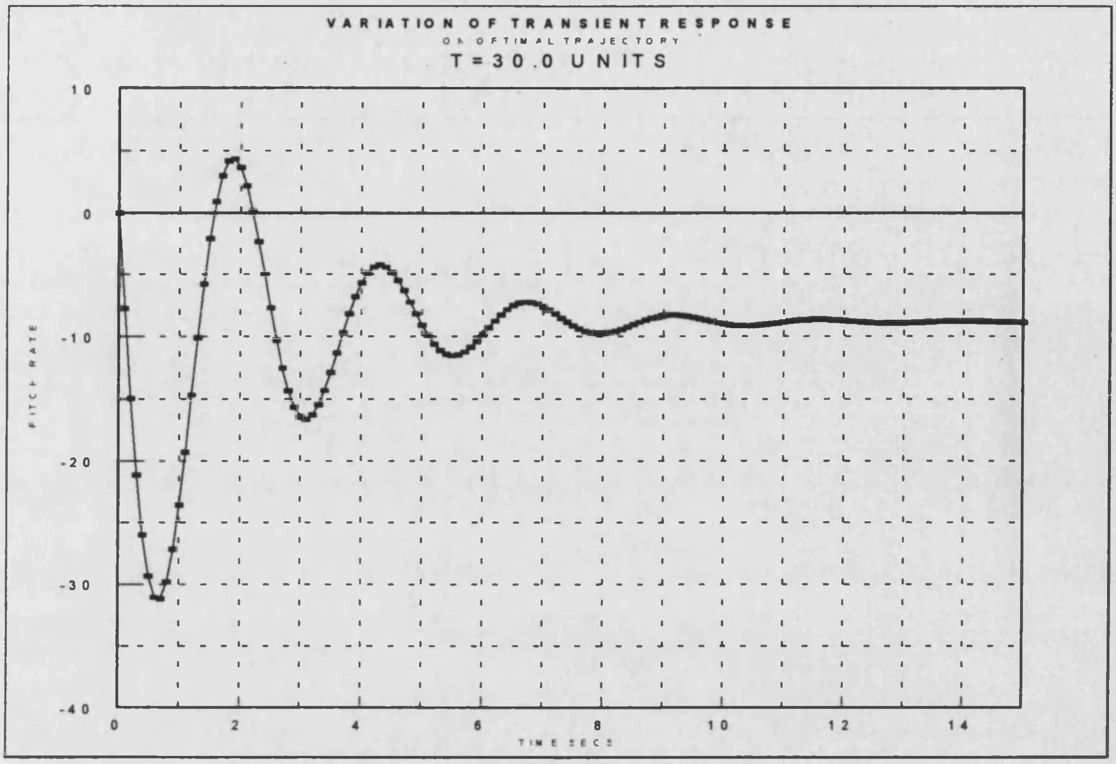

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T = 30.0 UNITS
10
0
-10
-20
-30
-40
RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 37

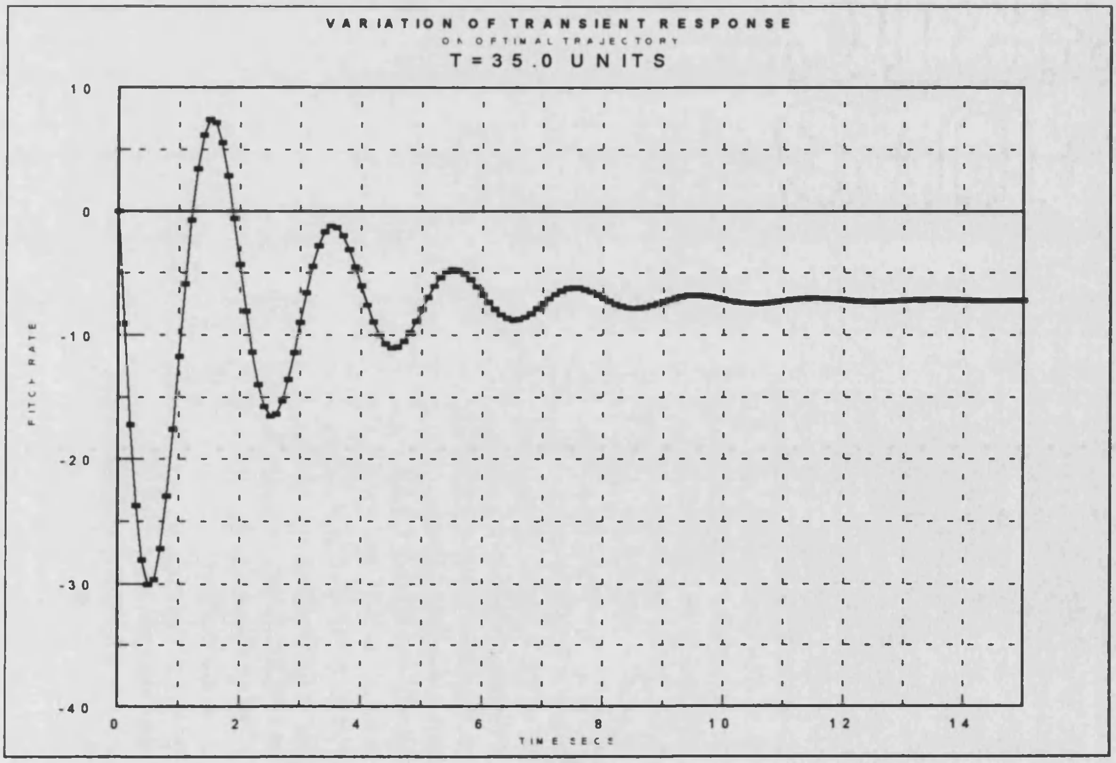

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T = 35.0 UNITS
10
0
-10
-20
-30
-40
RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 38

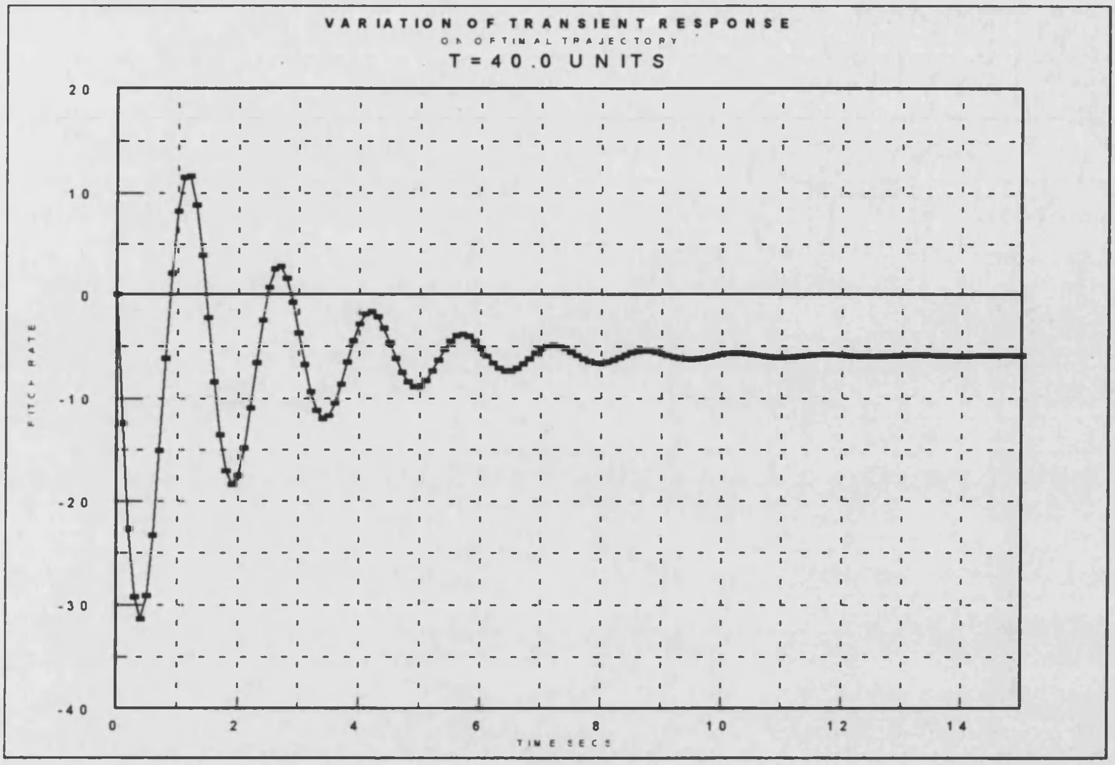

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T = 40.0 UNITS
20
10
0
-10
-20
-30
-40
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 39

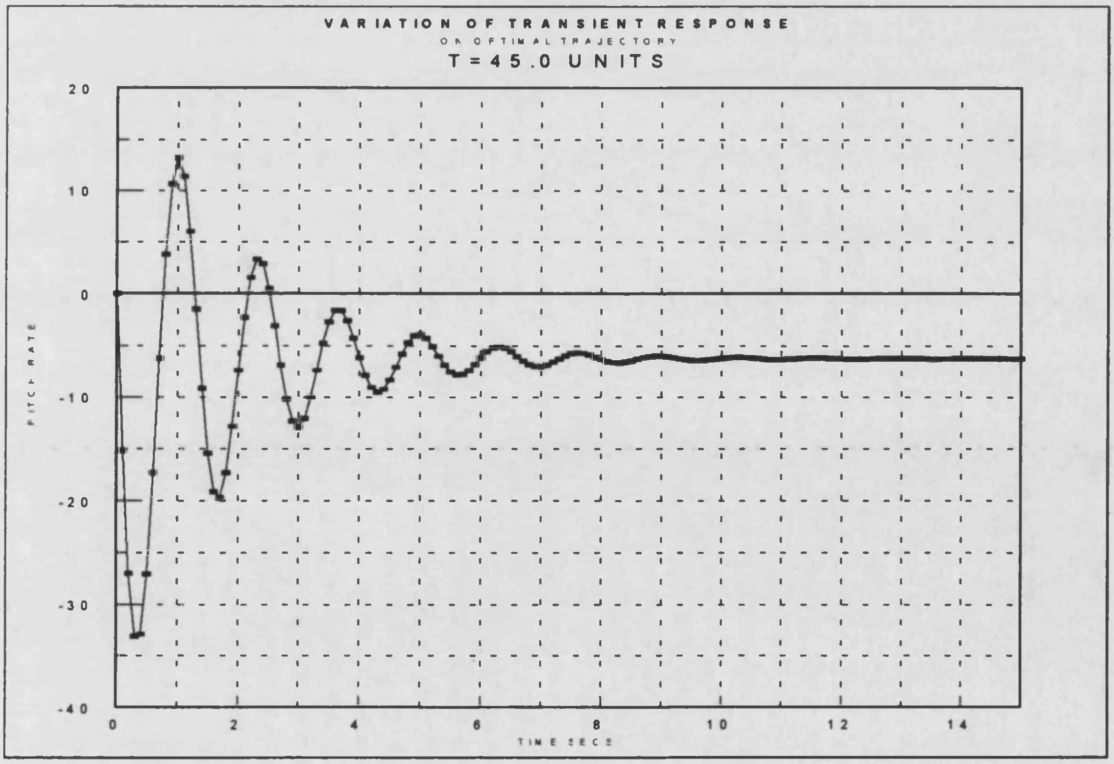

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T = 45.0 UNITS
20
10
0
-10
-20
-30
-40
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 40

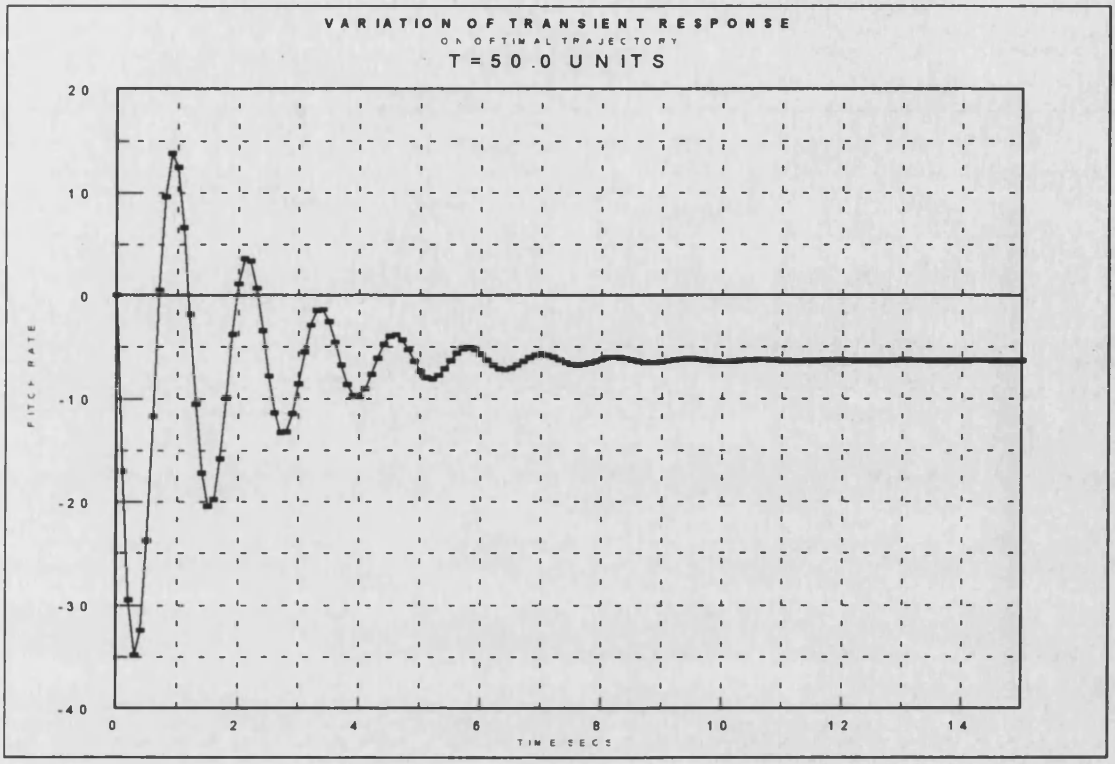

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T = 50.0 UNITS
20
10
0
-10
-20
-30
-40
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 41

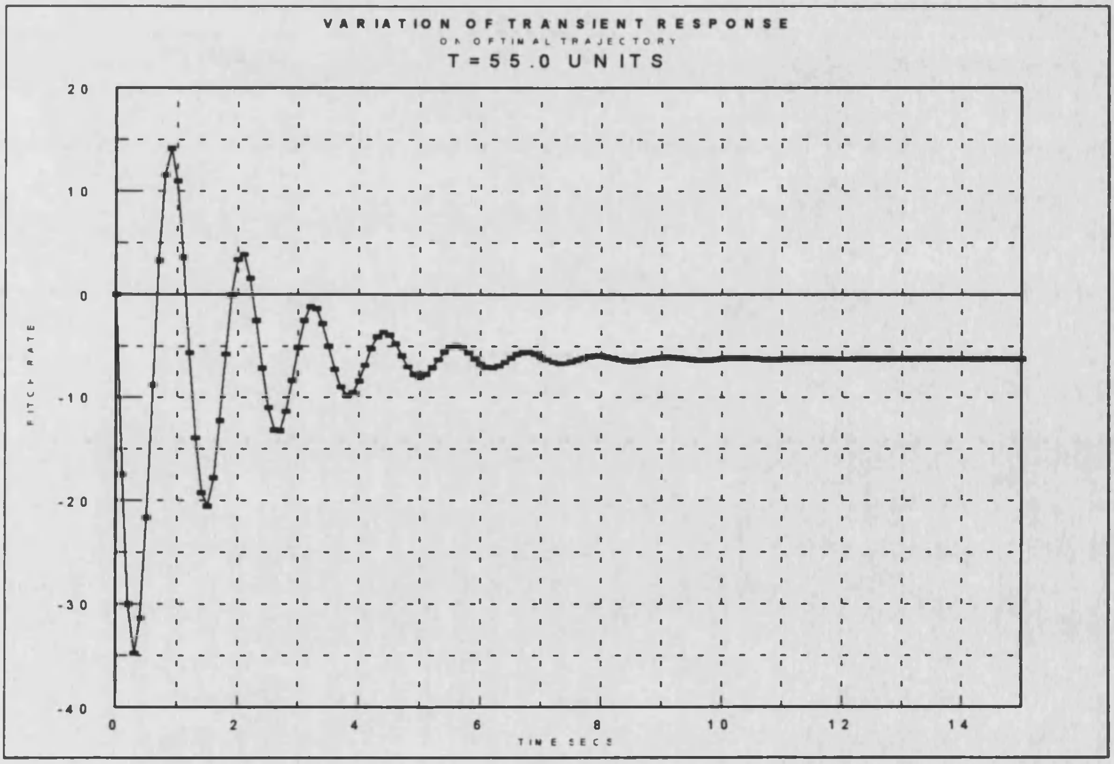
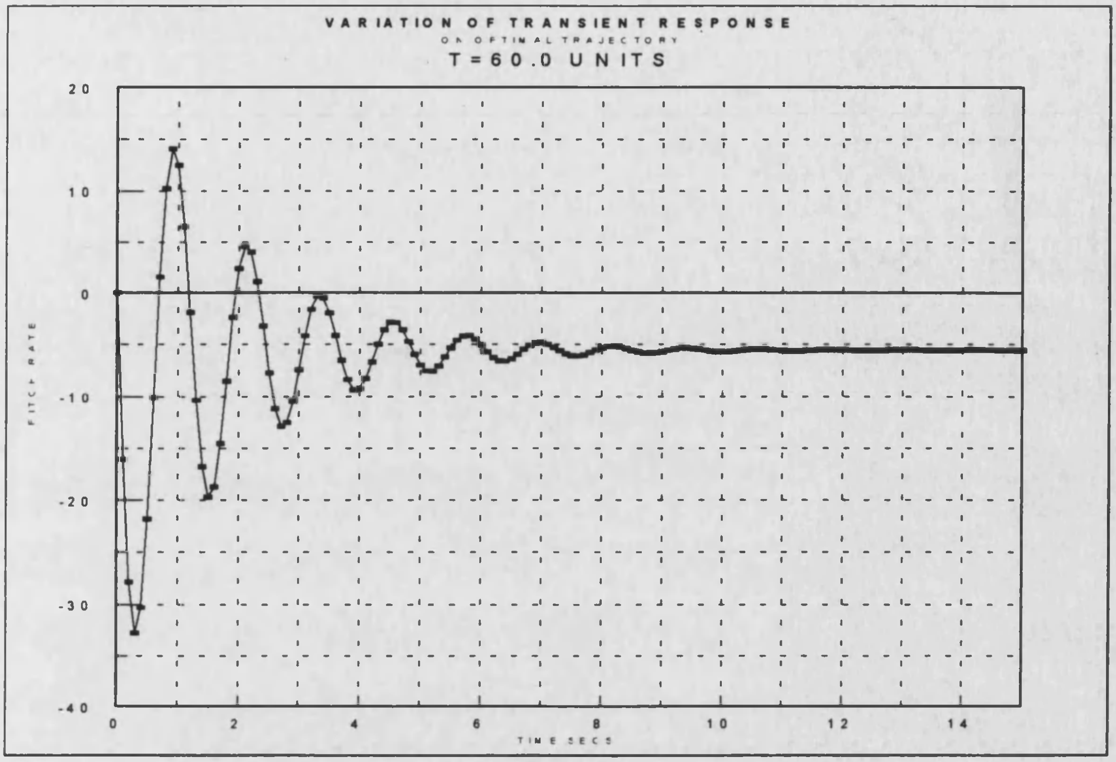
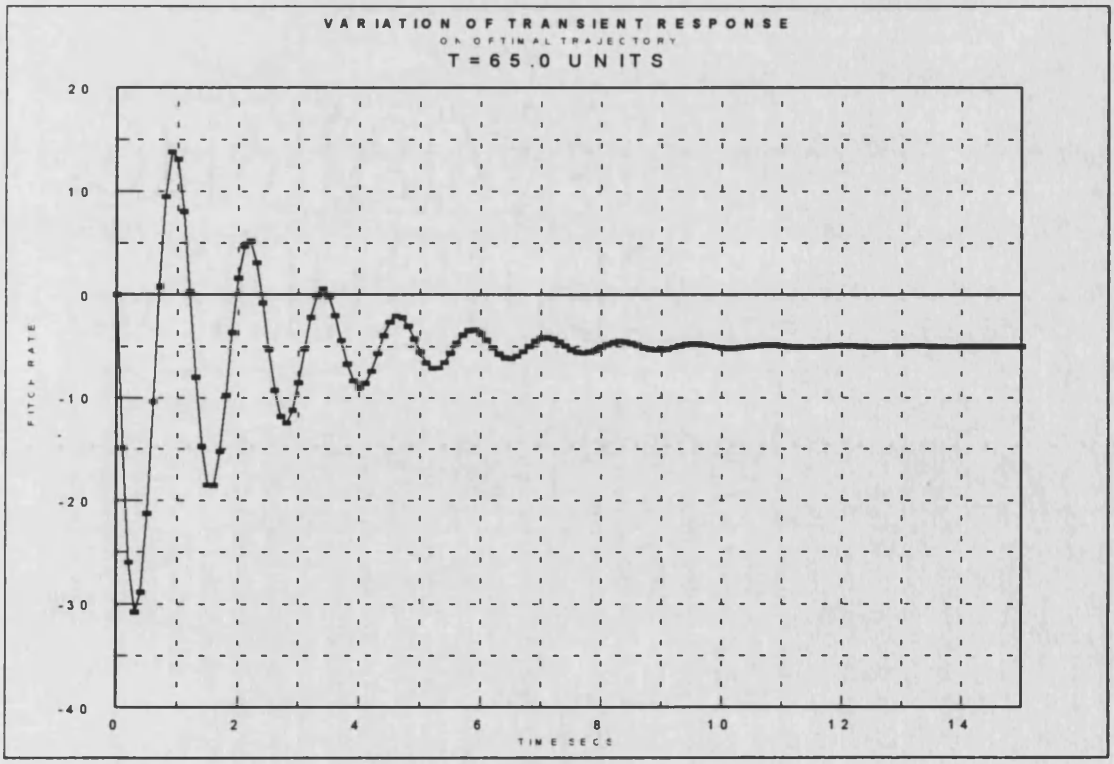

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T = 55.0 UNITS
20
10
0
-10
-20
-30
-40
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 42


VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T=60.0 UNITS
20
10
0
-10
-20
-30
-40
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS
Fig. 43


VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T=65.0 UNITS
20
10
0
-10
-20
-30
-40
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS
Fig. 44

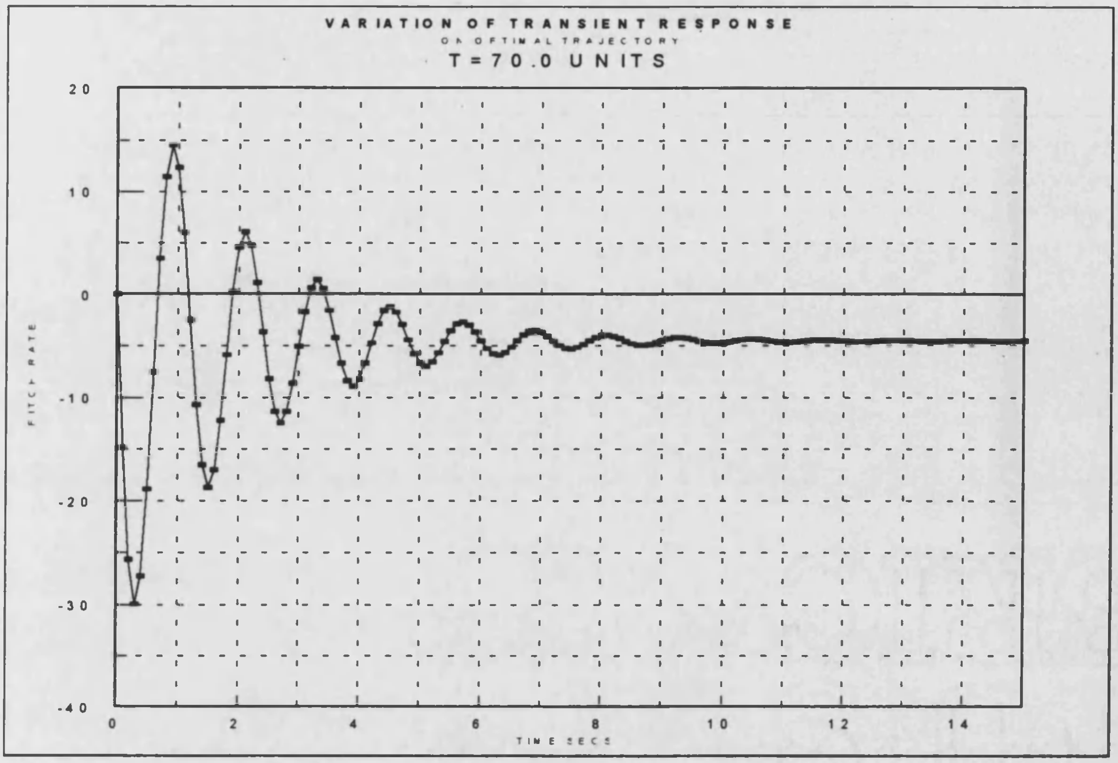

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T = 70.0 UNITS
20
10
0
-10
-20
-30
-40
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 45

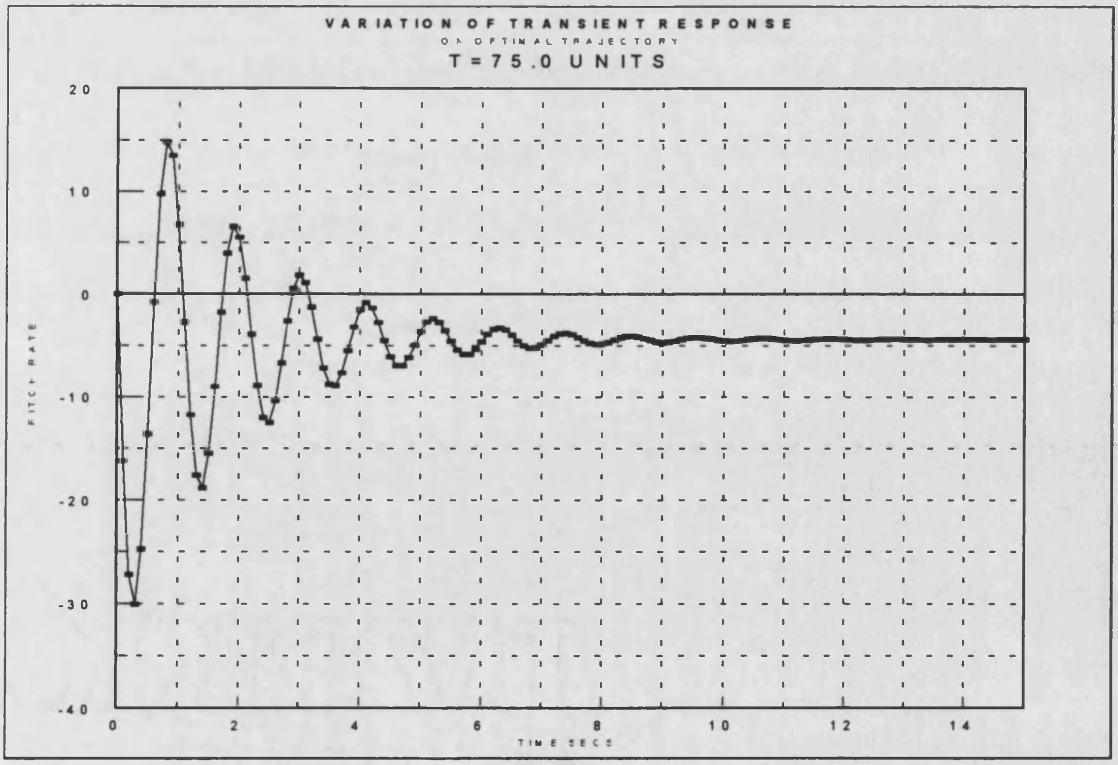

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T = 75.0 UNITS
20
10
0
-10
-20
-30
-40
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 46

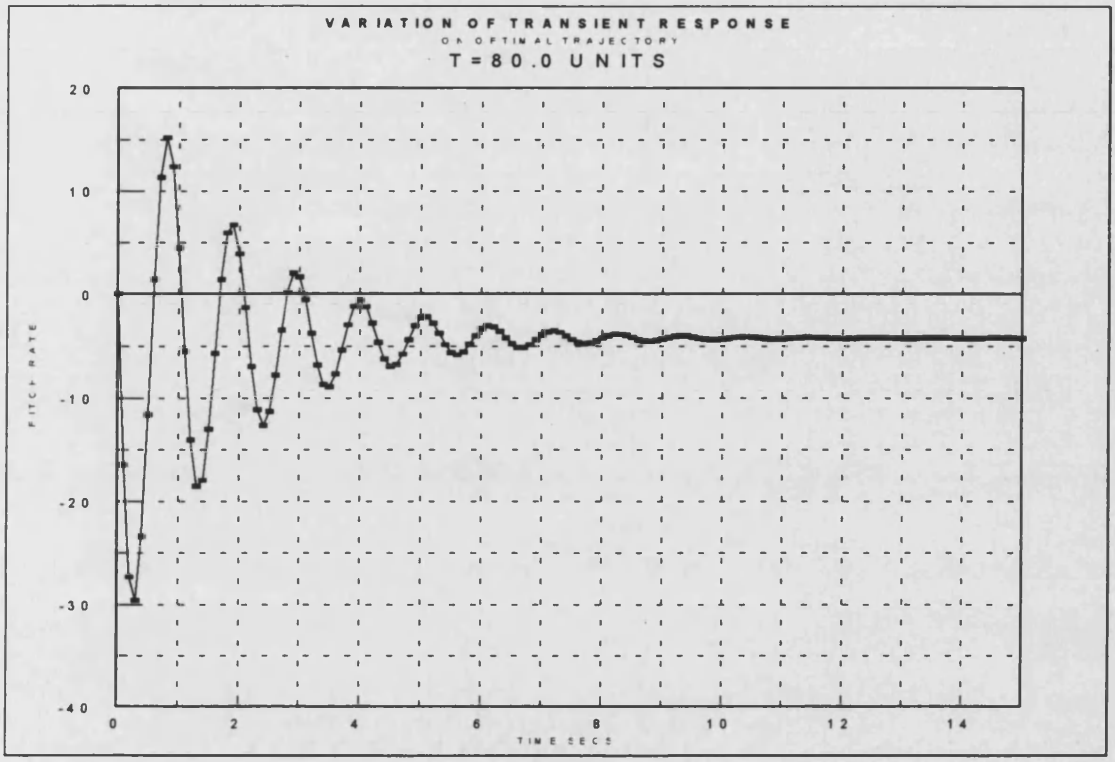

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T = 80.0 UNITS
20
10
0
-10
-20
-30
-40
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 47

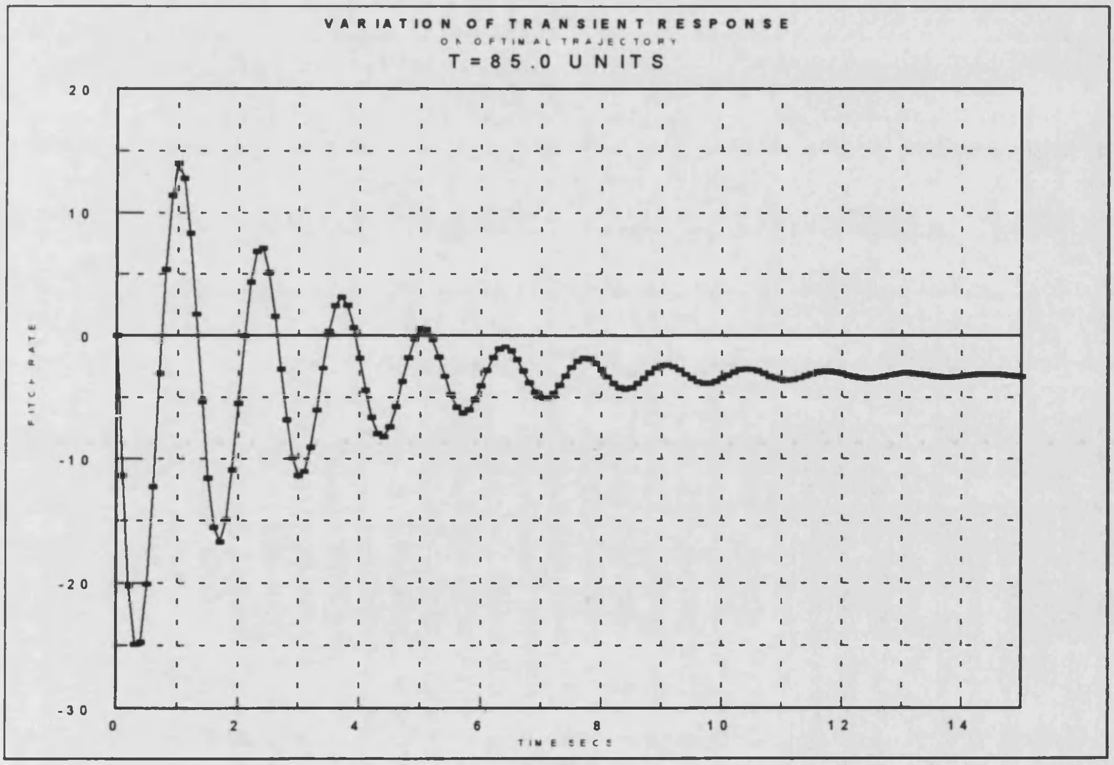

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T = 85.0 UNITS
20
10
0
-10
-20
-30
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 48

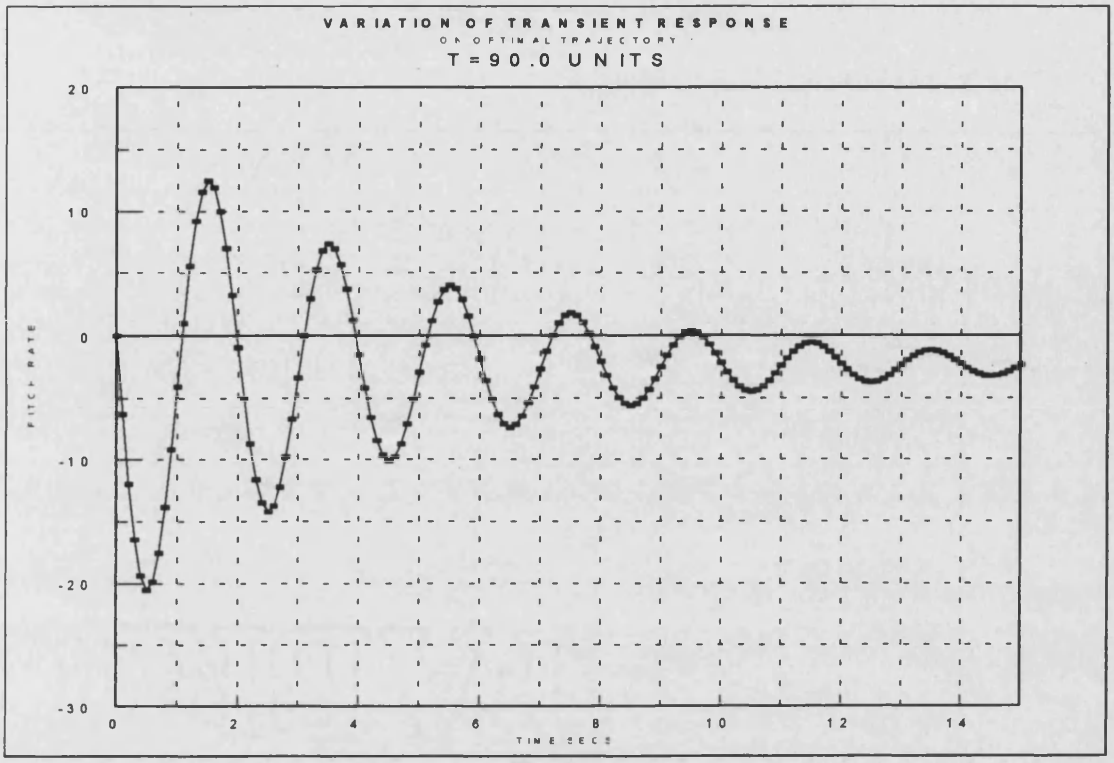

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T=90.0 UNITS
20
10
0
-10
-20
-30
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 49

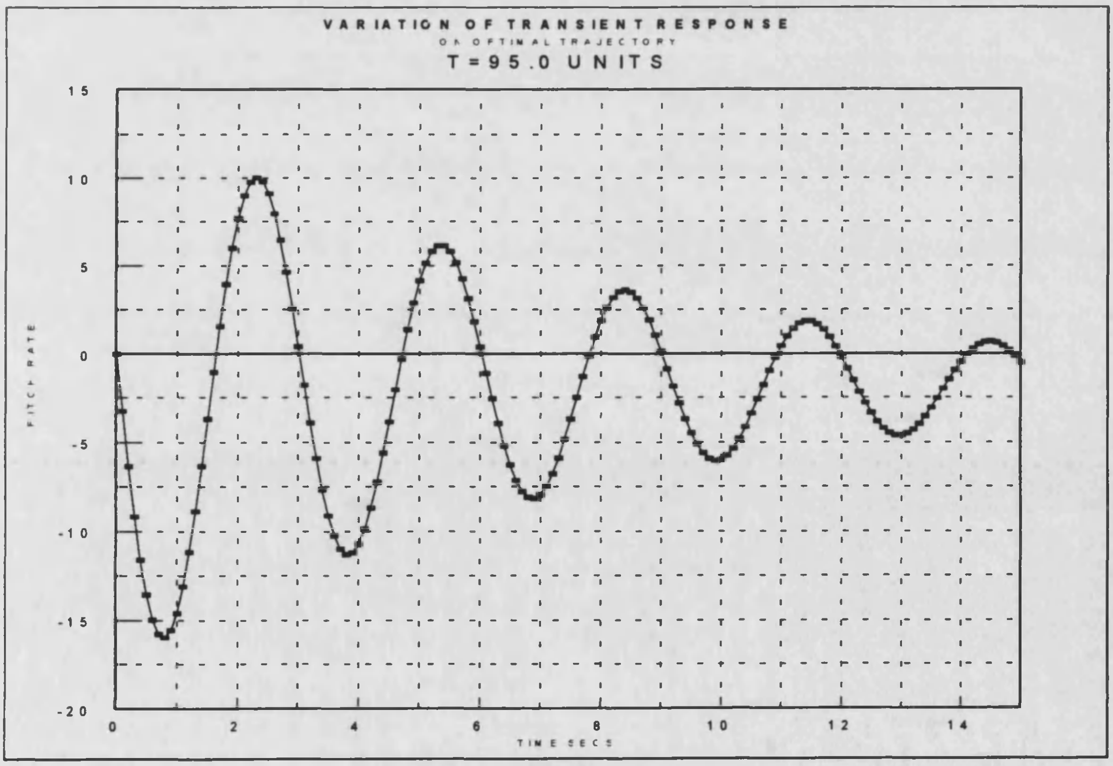
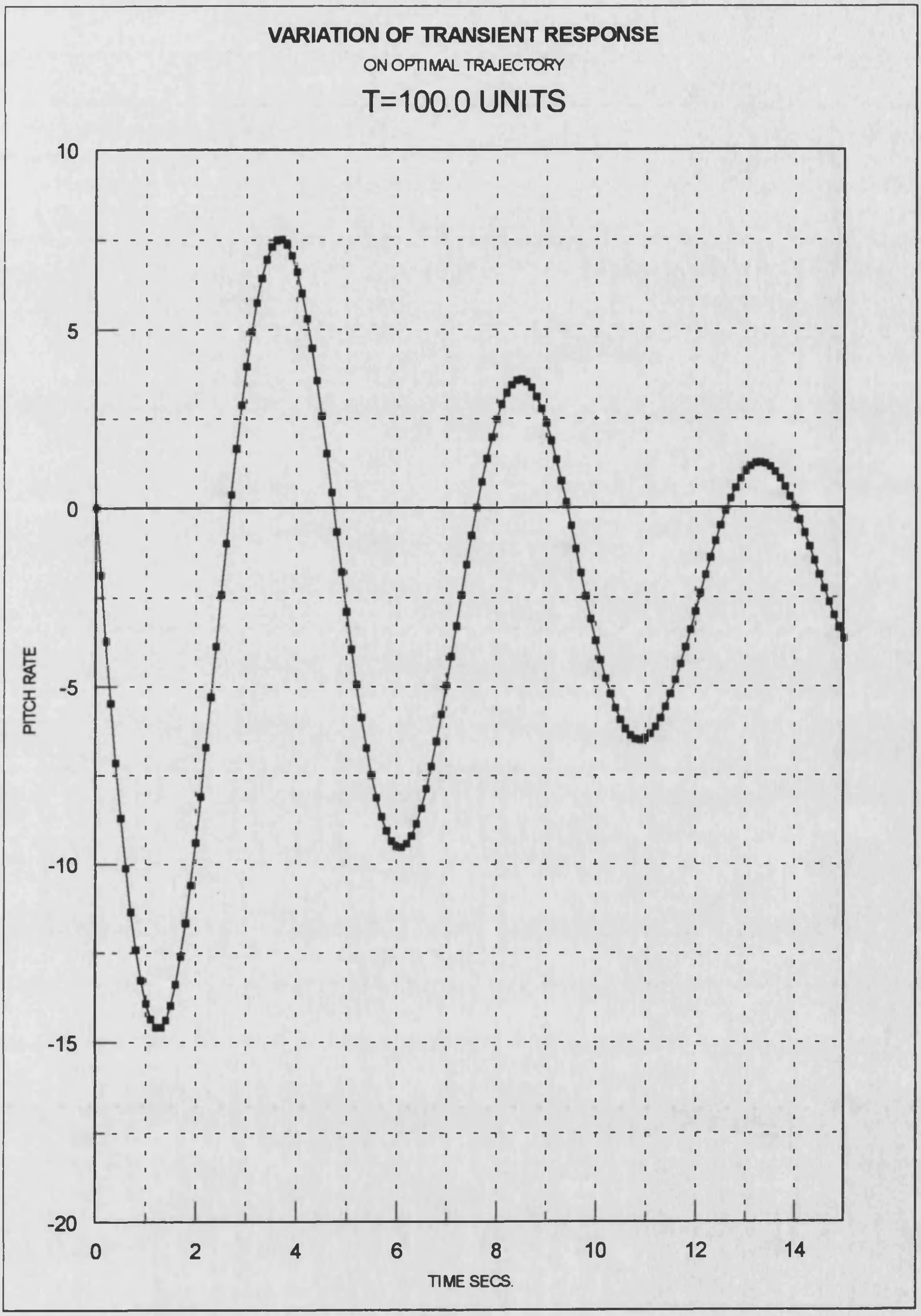

VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T=95.0 UNITS
15
10
5
0
-5
-10
-15
-20
PITCH RATE
0
2
4
6
8
10
12
14
TIME SECS

Fig. 50



VARIATION OF TRANSIENT RESPONSE
ON OPTIMAL TRAJECTORY
T=100.0 UNITS
PITCH RATE
TIME SECS.
10
5
0
-5
-10
-15
-20
0
2
4
6
8
10
12
14

Fig. 51

Chapter 6 gives a brief overview of adaptive control research. It describes the historical development of adaptive control from open-loop gain scheduling, through high-gain series model reference systems, parallel model reference systems, indirect adaptive control utilising an on-line idenification system, to on-line optimum adaptive control.

CHAPTER 6

ADAPTIVE CONTROL

Adaptive control has been the subject of dedicated research for over three decades. This chapter provides a brief overview of adaptive control techniques and the various approaches which have been considered in attempts at solving this problem. The overall objective of adaptive control is to produce a system which responds in the same manner even though the dynamics of the system being controlled may change as a function of environmental conditions. Specifically in the context of adaptive flight control, the dynamics of the aircraft being controlled change as a function of flight condition. The flight condition is generally defined as a function of altitude, speed, Mach number, dynamic pressure, or incidence. Historically the first attempts at adaptive control were based on controller gain scheduling techniques.

GAIN SCHEDULING:

This is one of the earliest approaches to adaptive control and it has been used extensively in the design of flight control systems since it was introduced in the late 1950s. and early 1960s. The method is based on determining a set of auxiliary variables, of the process being controlled, which relate to the changes occurring in the process dynamics. Once this relationship has been established it is then feasible to compensate for the changes in the system dynamics by rescheduling the controller parameters as a function of these auxiliary variables. A schematic representation of an open-loop gain scheduling controller is shown in figure 52.
Gain scheduling, while extensively used in the design of flight control systems is however an 'open-loop' form of adaptive system. If the dynamics of the plant being

controlled deviate from those anticipated to accrue at a specific set of auxiliary variables, the controller parameters will be set according to the rescheduling algorithm and not as a function of the actual plant parameters pertaining at that instant in time.

Gain scheduling has the added disadvantage that significant design time is required to generate the scheduling laws and to cater for all combinations of operating conditions of the process being controlled. With modern computational design techniques available this is no longer considered to be a serious restriction to the generation of rescheduling algorithms; nevertheless it is the desire to close the loop around the adaptive process that has motivated this research.

One significant advantage of gain scheduling is that controller parameters are computed directly as soon as the auxiliary variables are measured and can be set instantly without any inherent dynamics in the adaptive process itself. This in turn alleviates the stability problems frequently encountered in alternative adaptive systems.

Series Model Reference Adaptive Systems:-

Early design attempts to achieve closed-loop adaptive control were based on the so-called model reference scheme. Two forms of model reference methods have been researched. The first is the series model reference system which is the subject of this sub-section. The alternative approach is the parallel model reference scheme which is discussed in the next section of the thesis.

In model reference adaptive control the objective is to control the system such that the output response closely follows some desired response as defined by a system model. The system model chosen is generally of the same order as, or lower order

than, the actual system being controlled. A schematic diagram of a series model reference scheme is shown in figure 53.
In this scheme the series model response is used as the input to the closed-loop system of controller and plant. For the system to follow the model response accurately, the closed-loop transfer function of plant and controller must be maintained close to unity gain at all frequencies. To achieve this requirement the forward path open-loop gain of plant and controller must be maintained very high. This in turn is liable to lead to instability problems of the closed-loop system. A limit cycle detector was incorporated to determine the onset of oscillation and the forward path gain was reduced to re-stabilise the system. This technique seems contrary to the principles of good control system design where the objective is normally to retain good stability margins such that the system is never unstable. A significant disadvantage of this system is that oscillations are present in the system response. Additionally the high forward path gain can cause actuator saturation with catastrophic results. In 1966 an adaptive scheme similar to this was test flown on the Bell X-15 experimental aircraft. Control authority saturation in the pitch axis masked an instability in the roll axis and the aircraft exceeded its structural limitations resulting in the loss of the test vehicle.

PARALLEL MODEL REFERENCE ADAPTIVE SYSTEM:-

In this implementation the command signals are fed to the actual system and to a parallel model of the desired system response. The error between system and model responses is computed and used to adjust the parameters of either a forward path controller or a state feedback controller. It is also possible for the system to use a combination of forward path and feedback controllers. Early work on this type of system was undertaken by Whitaker in 1961. The system when working adjusts the controller parameters such that the output of the controlled process approaches the model response asymptotically. The update of the controller parameters was achieved using a gradient algorithm whereby the rate of change of the controller parameters was adjusted to be proportional to the sensitivity of output error with respect to controller parameter variations. The constant of proportionality is referred to as the adaptive gain of the system. The time derivative of the controller parameter is given by

$$\frac{dp}{dt} = -ge_0(p)\frac{\partial(y_s(p))}{\partial p}$$

where $\frac{\partial(y_s(p))}{\partial p}$ is the sensitivity function which depends on the unknown system parameters, g is the adaptive gain , and $e_0(p)$ is the error between the system response and the model response y_s and y_m respectively.

To overcome the fact that the sensitivity function depends on the unknown plant parameters M.I.T. developed a system which incorporated estimates of the plant parameters in the sensitivity function. This became known as the M.I.T. rule.

The stability analysis of this class of system is difficult; however for low values of adaptive gain and small forcing function amplitudes the system has been shown to be stable. Parks employed Lyapunov stability techniques to design model reference adaptive systems which could be proven to be asymptotically convergent to the model response and stable.

Hill-climbing techniques have also been utilised in an alternative approach to the design of model reference adaptive systems.

These methods unlike the gain scheduling method do not instantaneously adapt to the desired values of controller parameters and there is a finite adaptive response time. This can affect significantly, errors between the system response and some optimum desired model response. This is particularly true in the case where the controller and plant parameters are initially mismatched. In this case the system response can deviate significantly from the desired model response especially during transient responses, and it may take several repetitive applications of the command input before the adaptation is complete and the system response matches that of the desired model response. The speed of adaptation is dependent on the type of input command to the system as the error between system and model response is dependent on this. The speed of adaptation is also dependent on the choice of adaptive gain which is in turn dependent on stability considerations. The outcome of this is that the speed of adaptation is generally slower than the response time of the system and this results in significant deviations of the system transient response from the model optimum.

For many flight control applications this is unacceptable. An additional objective of this research then is to complete the adaptation of the controller well within the transient response time of the system and to augment the control signal to force the adapted system to minimise the error between the system response and the desired model response.

INDIRECT ADAPTIVE CONTROL:-

Indirect adaptive control generally incorporates an on-line identification scheme. The so-called self tuning controller is an example of such a scheme. This is depicted in figure 54. Model reference adaptive systems update the controller parameters without obtaining an explicit identification of the dynamics of the process to be controlled. By employing on-line identification of the plant certain advantages ensue. First of all, once the dynamics of the plant are known it is possible to incorporate an on-line redesign of the controller. The update of the controller parameters may merely be a rescheduling exercise based not on auxiliary variable measurements as in the open-loop adaptive controller but rather on the latest estimates of the actual process dynamics. Secondly, once the plant parameters are established, instead of merely adjusting the controller parameters to achieve some desired closed-loop transfer function, it is possible to re-compute the optimal control which will force the system response to minimise a function of the error between the actual system response and some desired nominal system response. When this is achieved within the transient response time of the system it can justifiably be referred to as an optimum adaptive controller. This then is a natural desirable extension of indirect adaptive control. Such a scheme was proposed by the author in ref. 25 and a block diagram of the scheme is reproduced in figure 55.

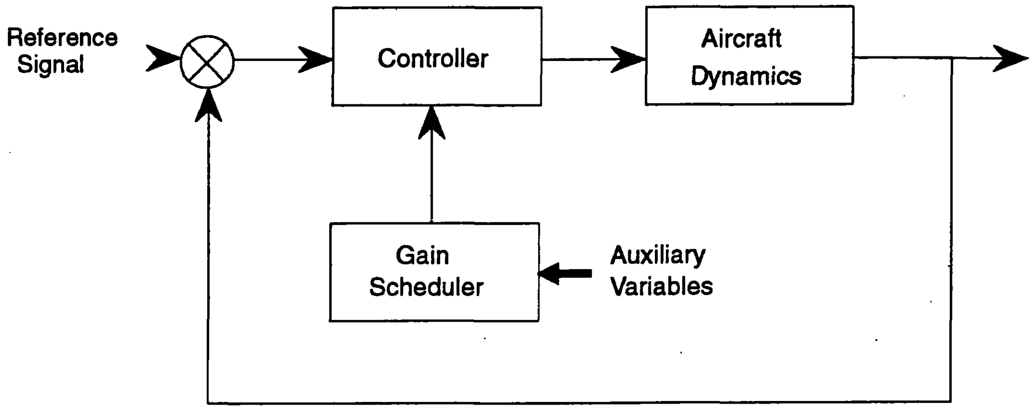
The on-line identification of system dynamics is often far from trivial, and much research effort has been applied to this task. The techniques employed range from frequency response methods, cross correlation techniques and even analysis of responses to various classes of excitation, e.g. step responses, ramp or exponential inputs. All of these techniques require additional test signals purely for the purpose of identification. These inputs in turn corrupt the system response usually in an undesirable fashion. The objective then is to produce an on-line identification scheme which uses only the normal operating control signals as the excitation of the system for identification purposes.

The Quasilinearisation algorithm referred to in the solution of two point boundary value problems resulting from the application of optimal control theory has been proposed by Bellman as a suitable technique for the on-line identification of systems.

Ref. 10. This method is suitable for the identification of slowly moving time variable parameters and has been selected as the technique to be employed in this investigation. In the case of the identification of flight dynamics the structure of the system dynamics is generally known and this greatly facilitates the identification process. Good estimates of both the control inputs and resultant system states are required in order to perform the system identification. If the system states cannot be measured because they are inaccessible they can generally be reconstructed by using asymptotic observer methods. The effects of noise, e.g. turbulence, measurement or structural may be alleviated by Kalman filtering. This is basically a time averaging method and it should be noted that this will slow up the speed of the identification algorithm. In an effort to get a thorough understanding of the deterministic identification of flight dynamics these difficulties have not been investigated as part of this thesis and remain an area for future investigative research. Given the availability of uncontaminated state and control a study has been undertaken which has demonstrated the continuous on-line tracking of system parameters and this is the subject of chapter 7.

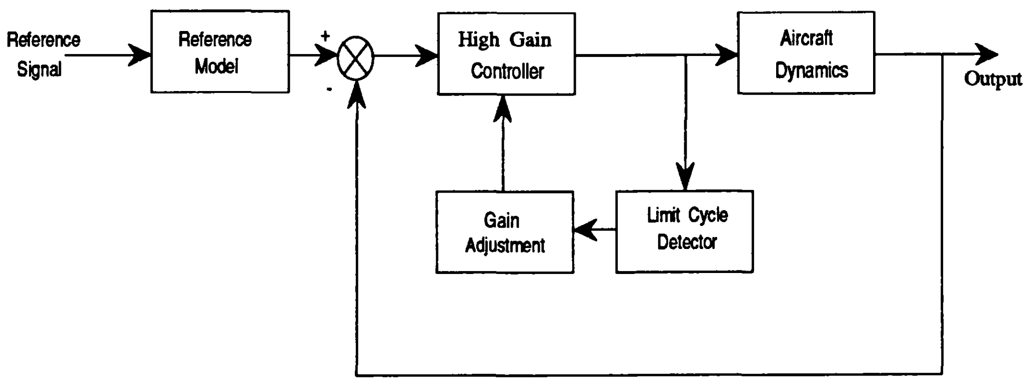
Kenneth and McGill have shown that within the convergence boundary of the Quasilinearisation algorithm, convergence quadratic. He has also indicated that if the system dynamics are linear then convergence is single step. This means that repeated iterative computations to identify the system, at the end of each identification interval, are only required if the system is non-linear. In the linear case the identification can proceed directly to the next identification interval. It should be noted that the identification process can be made continuous provided the system forcing function satisfies the persistent excitation requirements. For the system to be completely identified the forcing function is required to excite all of the modes of the system. In normal operation of the system this requirement on the nature of the forcing function will not always be satisfied. In this event, generally encountered as the system reaches steady state conditions, the identification cannot be performed. It is therefore necessary to inhibit any adaptive update of controller parameters if this condition is encountered. At or near steady state conditions of the system response this is not considered to be a significant problem.

The following chapter describes the design and performance of an on-line identification scheme in detail.



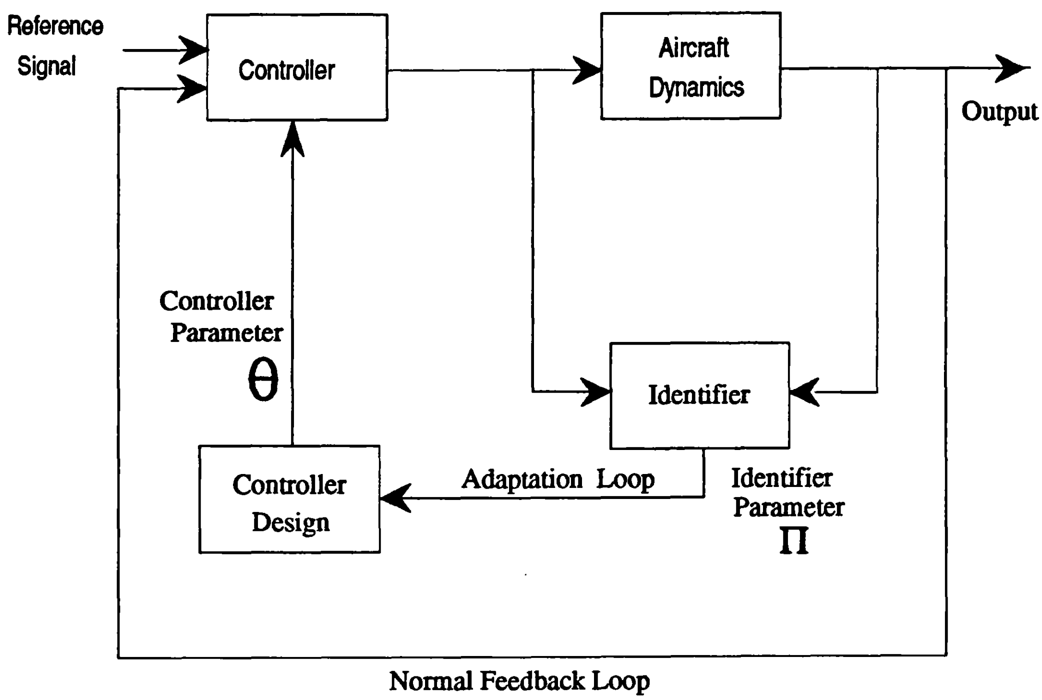
Open Loop Gain Scheduler

Fig. 52



Model Reference Adaptive Control - Series, High Gain Scheme

Fig. 53



Self Tuning Controller

Fig 54

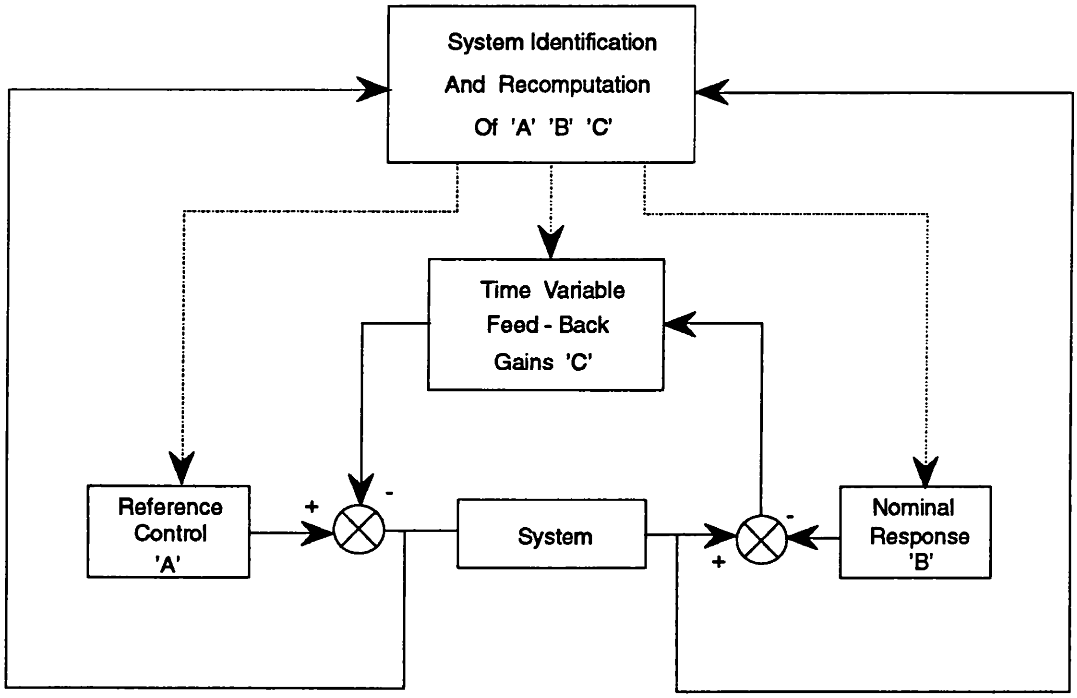


FIG. 55

Optimal Adaptive Controller

CHAPTER 7

ON-LINE IDENTIFICATION

This chapter is concerned with the continuous on-line identification of time varying non-linear system parameters.

A generalised system of this type can be defined as

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}, \underline{k}, t)$$

where

$\underline{x}$	represents the	n	dimensional	state	vector
$\underline{u}$	" "	m	"	control	"
$\underline{k}$	" "	i	"	parameter	"

The objective of the identification scheme is to produce a mathematical model representation of an unknown time varying system. This model when correct should produce ideally identical responses to the actual system when excited by the same forcing functions $\underline{u}$.

It should be noted that it is possible to introduce a degree of optimisation into the identification process. A typical performance index would be, for example, to minimise a function - normally quadratic - of state errors between the model and system. By choosing a weighting matrix on these state errors which is related to the inverse variance matrix of independent noise on the actual state variable measurements, the model can be made an optimum representation of the system.

For the purpose of this thesis, and to demonstrate the principles of on-line system identification, it is assumed that the system states and control inputs are deterministic and uncorrupted by measurement noise.

It is important to distinguish this identification model from the observation model of classical control which is used to reconstruct inaccessible states. To generate an observer model it is quite normal for the parameters of the system to be known. In the on-line identification model the system parameters are unknown and have to be identified.

Any identification process requires, as a starting point, a definition of the dimension of the mathematical model to represent the system. If the order of the system is unknown an iterative identification process can be performed whereby the dimension of the model is chosen and a "best fit" identification performed. The order of the model is then progressively increased until there is no significant improvement in the reduction of the error function between model and system.

In the case of aircraft dynamic identification this is unnecessary as both the dimension and structure of the system are well defined. This significantly simplifies the identification task.

Many different techniques for system identification have been investigated and these are well documented in ref. 30. One such technique is based on the Newton-Raphson algorithm. As this method has been utilised in the solution of two-point boundary value problems resulting from the application of the necessary conditions for optimal control, it is this approach which is used again for the purpose of on-line identification.

In the definition of the system dynamics the unknown system parameters are treated in a similar manner to the co-state variables in optimal control. As products of the states and parameters occur in the system definition the equations are essentially non-linear in character. The first step then is to linearise the system equations about a set of parameters. The linearisation algorithm in this case can be written as

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{k}} \end{bmatrix}_{n+1} = \left[J(\underline{x}_n;\underline{k}_n)\right]\begin{bmatrix} \underline{x}_{n+1} - \underline{x}_n \\ \underline{k}_{n+1} - \underline{k}_n \end{bmatrix} + f(\underline{x}_n,\underline{u}_n,\underline{k}_n,t)$$

Although the actual system parameters are time varying, it is assumed for the purpose of identification that they are constant during the short identification intervals. At the end of each identification the best constant values for the parameter set is obtained. The identification process is on-going and in this manner the time varying parameters are tracked as piece-wise constant values. The result is similar to a discrete sampling of the time varying parameters.

To illustrate this technique the particular case of identifying the pitch short-period small perturbation dynamic representation of an aircraft is considered. This representation is given by:-

$$\frac{q}{\eta} = \frac{K_0\omega_n^2(1+sT)}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

In matrix control canonical form this becomes

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ K_0\omega_n^2 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & T \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

In this representation the output is a function of both states and it is helpful to transform this to an observer representation for the purpose of identification. Redefining the system states, the equations become

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -k_1 & 1 \\ -k_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} k_3 \\ k_4 \end{bmatrix} u$$

$$y = x_1$$

where the output y and system state x_1 are the pitch rate of the aircraft, u is the elevator control input η, and the system parameters to be identified are the

$$k_i \;---\; i = 1,2,3,4$$

Specifically:-

$$\begin{aligned}
k_1 &= 2\xi\omega_n \\
k_2 &= \omega_n^2 \\
k_3 &= K_0\omega_n^2 T \\
k_4 &= K_0\omega_n^2
\end{aligned}$$

A block schematic diagram of this system representation is given in fig. 56.

Treating these parameters in a similar fashion to the co-states in the optimal control study, the linearised equations become

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{k}_1 \\ \dot{k}_2 \\ \dot{k}_3 \\ \dot{k}_4 \end{bmatrix}_{N+1} = \begin{bmatrix} -k_{1N} & 1 & -x_{1N} & 0 & u_N & 0 \\ -k_{2N} & 0 & 0 & -x_{1N} & 0 & u_N \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{1N+1} - x_{1N} \\ x_{2N+1} - x_{2N} \\ k_{1N+1} - k_{1N} \\ k_{2N+1} - k_{2N} \\ k_{3N+1} - k_{3N} \\ k_{4N+1} - k_{4N} \end{bmatrix} + \begin{bmatrix} -k_{1N}x_{1N} + x_{2N} + k_{3N}u_N \\ -k_{2N}x_{1N} + k_{2N}u_N \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The parameter identification task is now a boundary value problem in that the initial conditions on the unknown parameters have to be chosen such that the model states are satisfied at several points in time. Because there are more unknown parameters than there are states, in this instance twice as many, two points in time are chosen at which the model states must be matched to the actual system states. These time points have been chosen as the mid-point and end-point of the identification interval, in this instance. If only one state is accessible for measurement then the identification can still be performed by matching that state for both the system and model at four points in time.

The procedure for the solution of the multi-point boundary value problem for the above set of linearised equations is the same as that used for the solution of the optimal control boundary value problem. The procedure commences with selecting a set of starting vectors in time for the coefficients of the Jacobian matrix. The obvious choice for initialising the N$^{th.}$ iteration is to choose the actual system states and system control input of the system to be identified. The starting vectors for the four unknown parameters are chosen such that $k_{iN}(t) = g_i \quad i = 1;--;4$

Since the time derivatives of the unknown parameters have been assumed to be zero, the g_i are just four constants. If the range of the unknown parameters can be determined, an appropriate starting point would be to choose to set each parameter to the mid-point of its range. The initial conditions of each of the k_{iN+1} parameters must also be chosen, and since there is no specific preference for this they can be made equal to the last iteration values. Hence $k_{iN+1} = g_i \quad i = 1;--4$ is chosen.

Expanding the linearised equations and making the above substitutions, the linearised equations become:-

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{k}_1 \\ \dot{k}_2 \\ \dot{k}_3 \\ \dot{k}_4 \end{bmatrix}_{N+1} = \begin{bmatrix} -g_1 & 1 & -x_{1s} & 0 & u_s & 0 \\ -g_2 & 0 & 0 & -x_{1s} & 0 & u_s \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{1N+1} \\ x_{2N+1} \\ k_{1N+1} \\ k_{2N+1} \\ k_{3N+1} \\ k_{4N+1} \end{bmatrix} + \begin{bmatrix} g_1 x_{1s} \\ g_2 x_{1s} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The solution of these equations consists of a "Particular Integration" obtained by integrating the system of equations with initial conditions on the unknown parameters as $k_i(0) = g_i$; together with a linear combination of four sets of homogeneous solutions.

The complete solution of these equations is given by:

$$\begin{bmatrix} x_{1m}(t) \\ x_{2m}(t) \\ k_1(t) \\ k_2(t) \\ k_3(t) \\ k_4(t) \end{bmatrix} = C_1 \begin{bmatrix} x_{1H1}(t) \\ x_{2H1}(t) \\ k_{1H1}(t) \\ k_{2H1}(t) \\ k_{3H1}(t) \\ k_{4H1}(t) \end{bmatrix} + C_2 \begin{bmatrix} x_{1H2}(t) \\ x_{2H2}(t) \\ k_{1H2}(t) \\ k_{2H2}(t) \\ k_{3H2}(t) \\ k_{4H2}(t) \end{bmatrix} + C_3 \begin{bmatrix} x_{1H3}(t) \\ x_{2H3}(t) \\ k_{1H3}(t) \\ k_{2H3}(t) \\ k_{3H3}(t) \\ k_{4H3}(t) \end{bmatrix} + C_4 \begin{bmatrix} x_{1H4}(t) \\ x_{2H4}(t) \\ k_{1H4}(t) \\ k_{2H4}(t) \\ k_{3H4}(t) \\ k_{4H4}(t) \end{bmatrix} + \begin{bmatrix} x_{1P.I.}(t) \\ x_{2P.I.}(t) \\ k_{1P.I.}(t) \\ k_{2P.I.}(t) \\ k_{3P.I.}(t) \\ k_{4P.I.}(t) \end{bmatrix}$$

Eqn. 2

It should be noted that these equations are valid throughout the identification interval. The constants C_i are weighting factors on the homogeneous integrations to correct the particular integrations to give the desired values of the system model time response and the identified values of the unknown parameters k_i These constants must be determined from the state responses of the model. Since there are four constants to be evaluated and only two states, the above equations are applied at the mid-point and end-point of the identification interval. i.e. at times t_1 and t_f respectively.

With the above set of initial conditions for the unknown parameters the particular integration reduces to:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}_{P.I.} = \begin{bmatrix} -g_1 & 1 \\ -g_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{P.I.} + \begin{bmatrix} g_3 \\ g_4 \end{bmatrix} u_s$$

It should be noted that the particular integration system defined above is a model of the original system having an identical structure but with estimates chosen for the unknown parameters of the actual system. As the choice of the parameters for this system model will initially be incorrect, so the model responses given by the particular integration will not match the true system responses and these are corrected by the linear combination of the homogeneous integrations.

There are four sets of homogeneous integrations required because there are four unknown initial conditions on the parameters. The initial conditions for each set of homogeneous solution are chosen as follows:-

$$\begin{bmatrix} x_{1Hi} \\ x_{2Hi} \\ k_{1Hi} \\ k_{2Hi} \\ k_{3Hi} \\ k_{4Hi} \end{bmatrix}_{t=0} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}_{i=1} ; \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}_{i=2} ; \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}_{i=3} ; \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}_{i=4}$$

With these sets of initial conditions for the homogeneous integrations, the set of four homogenous integrations become respectively:-

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}_{H1} = \begin{bmatrix} -g_1 & 1 \\ -g_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{H1} + \begin{bmatrix} -x_{1s} \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}_{H2} = \begin{bmatrix} -g_1 & 1 \\ -g_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{H2} + \begin{bmatrix} 0 \\ -x_{1s} \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}_{H3} = \begin{bmatrix} -g_1 & 1 \\ -g_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{H3} + \begin{bmatrix} u_s \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}_{H4} = \begin{bmatrix} -g_1 & 1 \\ -g_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{H4} + \begin{bmatrix} 0 \\ u_s \end{bmatrix}$$

The Particular Integral and the four sets of homogeneous integrations have all to be integrated simultaneously and if the simulation of the original system is included this results in a total of twelve differential equations to be integrated. This complete set of differential equations is given below in matrix form as:

$$\begin{bmatrix} \dot{x}_{1s} \\ \dot{x}_{2s} \\ \dot{x}_{1P.I.} \\ \dot{x}_{2P.I.} \\ \dot{x}_{1H1} \\ \dot{x}_{2H1} \\ \dot{x}_{1H2} \\ \dot{x}_{2H2} \\ \dot{x}_{1H3} \\ \dot{x}_{2H3} \\ \dot{x}_{1H4} \\ \dot{x}_{2H4} \end{bmatrix} = \begin{bmatrix} -k_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -k_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -g_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -g_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -g_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -g_2 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & -g_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -g_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g_1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g_1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g_2 & 0 \end{bmatrix} \begin{bmatrix} x_{1s} \\ x_{2s} \\ x_{1P.I.} \\ x_{2P.I.} \\ x_{1H1} \\ x_{2H1} \\ x_{1H2} \\ x_{2H2} \\ x_{1H3} \\ x_{2H3} \\ x_{1H4} \\ x_{2H4} \end{bmatrix} + \begin{bmatrix} k_3 u_s \\ k_4 u_s \\ g_3 u_s \\ g_4 u_s \\ -x_{1s} \\ 0 \\ 0 \\ -x_{1s} \\ u_s \\ 0 \\ 0 \\ u_s \end{bmatrix}$$

To solve for the four constants C_i in equation (2) the desired state responses of the model are set to those of the actual system to be identified. The initial conditions on the model particular integration are set equal to the value of the actual system states at the start of the identification interval. The following set of equations then apply:

$$\begin{bmatrix} x_{1s}(t_0) \\ x_{2s}(t_0) \\ x_{1s}(t_1) \\ x_{2s}(t_1) \\ x_{1s}(t_f) \\ x_{2s}(t_f) \end{bmatrix} = C_1 \begin{bmatrix} x_{1H1}(t_0) \\ x_{2H1}(t_0) \\ x_{1H1}(t_1) \\ x_{2H1}(t_1) \\ x_{1H1}(t_f) \\ x_{2H1}(t_f) \end{bmatrix} + C_2 \begin{bmatrix} x_{1H2}(t_0) \\ x_{2H2}(t_0) \\ x_{1H2}(t_1) \\ x_{2H2}(t_1) \\ x_{1H2}(t_f) \\ x_{2H2}(t_f) \end{bmatrix} + C_3 \begin{bmatrix} x_{1H3}(t_0) \\ x_{2H3}(t_0) \\ x_{1H3}(t_1) \\ x_{2H3}(t_1) \\ x_{1H3}(t_f) \\ x_{2H3}(t_f) \end{bmatrix} + C_4 \begin{bmatrix} x_{1H4}(t_0) \\ x_{2H4}(t_0) \\ x_{1H4}(t_1) \\ x_{2H4}(t_1) \\ x_{1H4}(t_f) \\ x_{2H4}(t_f) \end{bmatrix} + \begin{bmatrix} x_{1p.I.}(t_0) \\ x_{2P.I.}(t_0) \\ x_{1P.I.}(t_1) \\ x_{2P.I.}(t_1) \\ x_{1P.I.}(t_f) \\ x_{2P.I.}(t_f) \end{bmatrix}$$

Subtracting the equations at t_0 from those at t_1 and t_f for each of the respective states and noting that $x_{ip.I.}(t_0) = x_{is}(t_0)$ and $x_{iHj}(t_0) = 0 \quad i = 1,2 \ ; j = 1,4$

the equations can be rearranged to give the solution of the four constants as

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix} = \begin{bmatrix} x_{1H1}(t_1) & x_{1H2}(t_1) & x_{1H3}(t_1) & x_{1H4}(t_1) \\ x_{2H1}(t_1) & x_{2H2}(t_1) & x_{2H3}(t_1) & x_{2H4}(t_1) \\ x_{1H1}(t_f) & x_{1H2}(t_f) & x_{1H3}(t_f) & x_{1H4}(t_f) \\ x_{2H1}(t_f) & x_{2H2}(t_f) & x_{2H3}(t_f) & x_{2H4}(t_f) \end{bmatrix}^{-1} \begin{bmatrix} x_{1s}(t_1) - x_{1P.I}(t_1) \\ x_{2s}(t_1) - x_{2P.I}(t_1) \\ x_{1s}(t_f) - x_{1P.I}(t_f) \\ x_{2s}(t_f) - x_{2P.I}(t_f) \end{bmatrix}$$

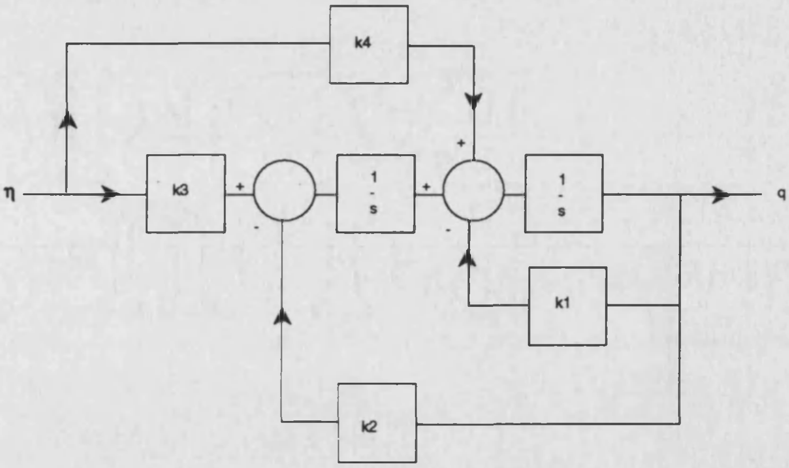
The corrected or identified values for the unknown parameters during the identification interval then become

$$\begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{bmatrix} = [I] \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix}$$

The identified values of the system parameters so determined then become the estimates of the unknown parameters for the next identification interval, to enable the next particular integration and next sets of homogeneous integrations to be performed. In this manner the identification process is continuous and enables time-varying parameters to be tracked as constants over each identification interval. By testing the matrix of homogeneous solutions to determine that it is non-singular it can be established that the identification is valid or otherwise at each step. A schematic diagram of the identification process is shown in fig. 57.

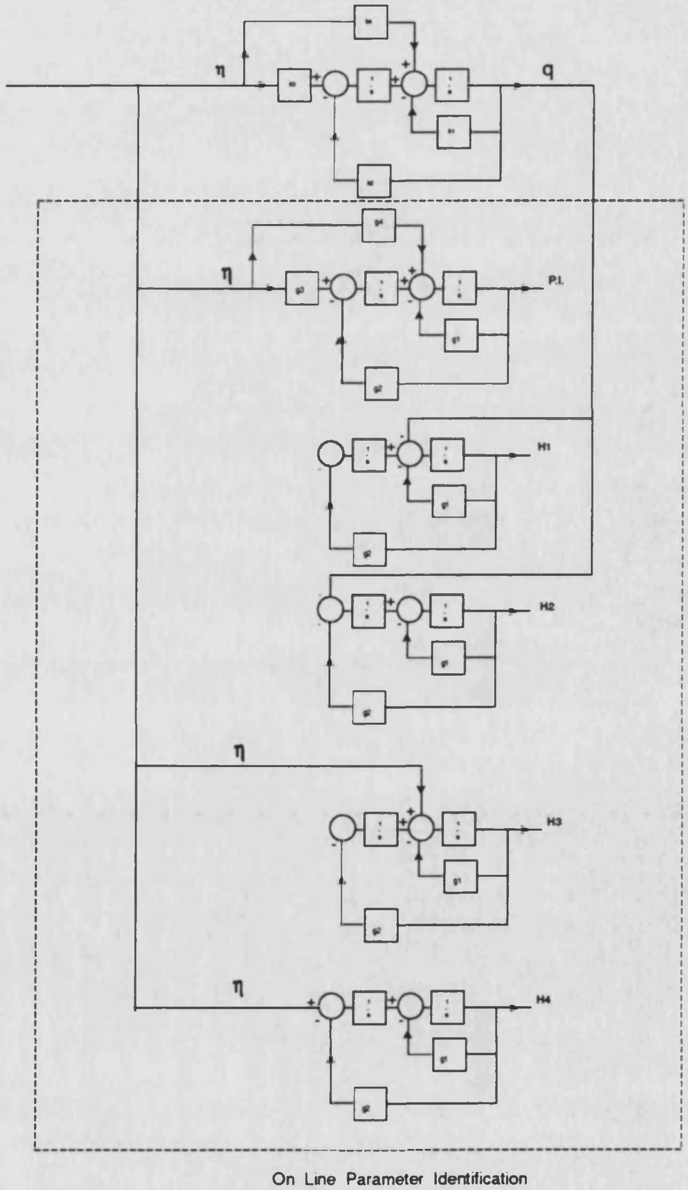
The results of the on-line identification and tracking of the aircraft system parameters by the Quasilinearisation technique are presented in figs. 58-61. The accuracy of the method is clearly demonstrated.

Chapter 8 defines a command stability augmentation system which adapts the controller parameters as a function of the on-line identified aircraft parameters.



Aircraft Pitch Loop For Identification

Fig.56



Identification Procedure

Fig. 57

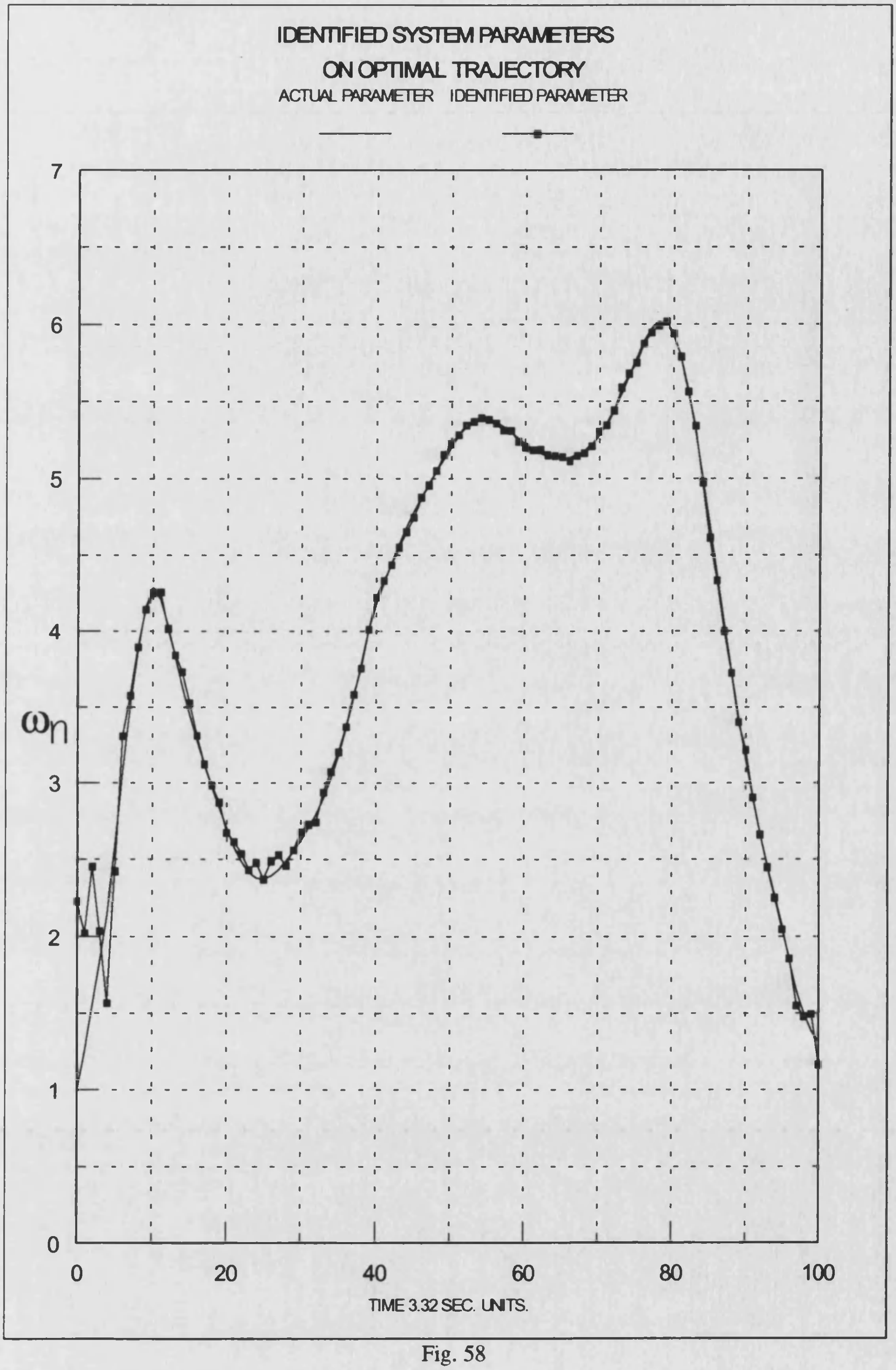

IDENTIFIED SYSTEM PARAMETERS
ON OPTIMAL TRAJECTORY
ACTUAL PARAMETER
IDENTIFIED PARAMETER
ω_η
0
1
2
3
4
5
6
7
0
20
40
60
80
100
TIME 3.32 SEC. UNITS.

Fig. 58

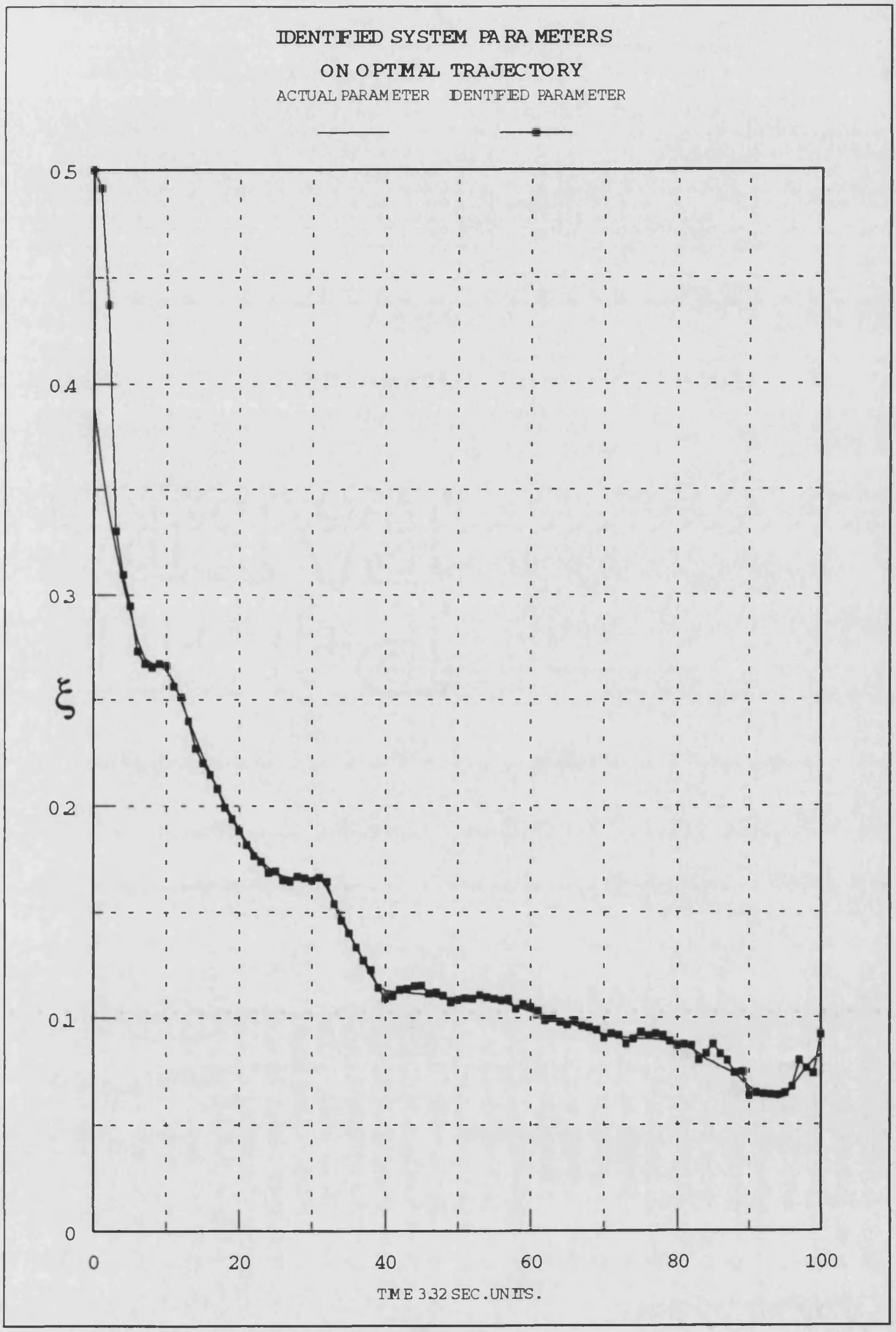

IDENTIFIED SYSTEM PARAMETERS
ON OPTIMAL TRAJECTORY
ACTUAL PARAMETER
IDENTIFIED PARAMETER
0.5
0.4
0.3
0.2
0.1
0
ξ
0
20
40
60
80
100
TIME 3.32 SEC. UNITS.

Fig. 59

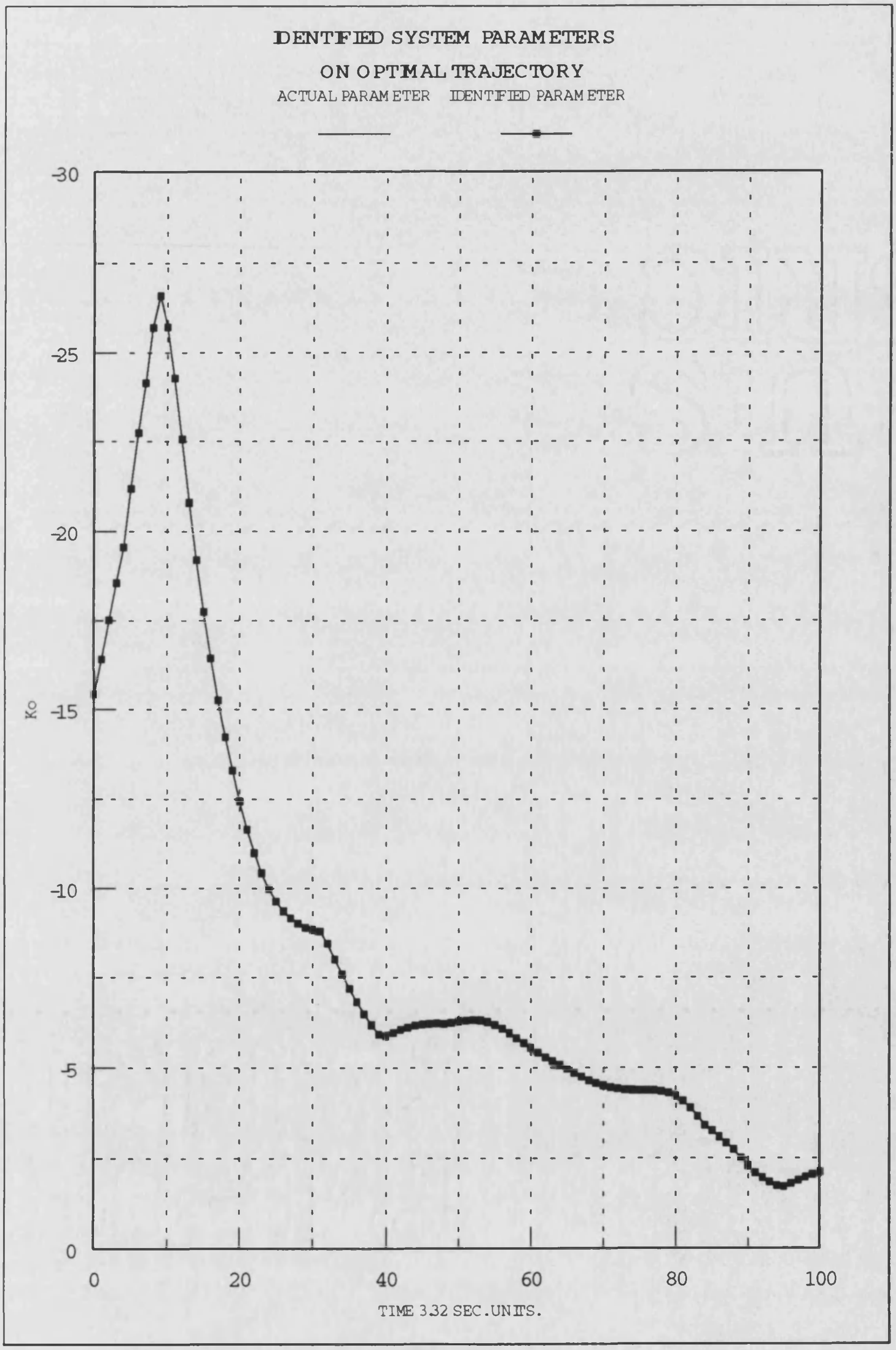

IDENTIFIED SYSTEM PARAMETERS
ON OPTIMAL TRAJECTORY
ACTUAL PARAMETER
IDENTIFIED PARAMETER
-30
-25
-20
-15
-10
-5
0
Ko
0
20
40
60
80
100
TIME 3.32 SEC.UNITS.

Fig. 60

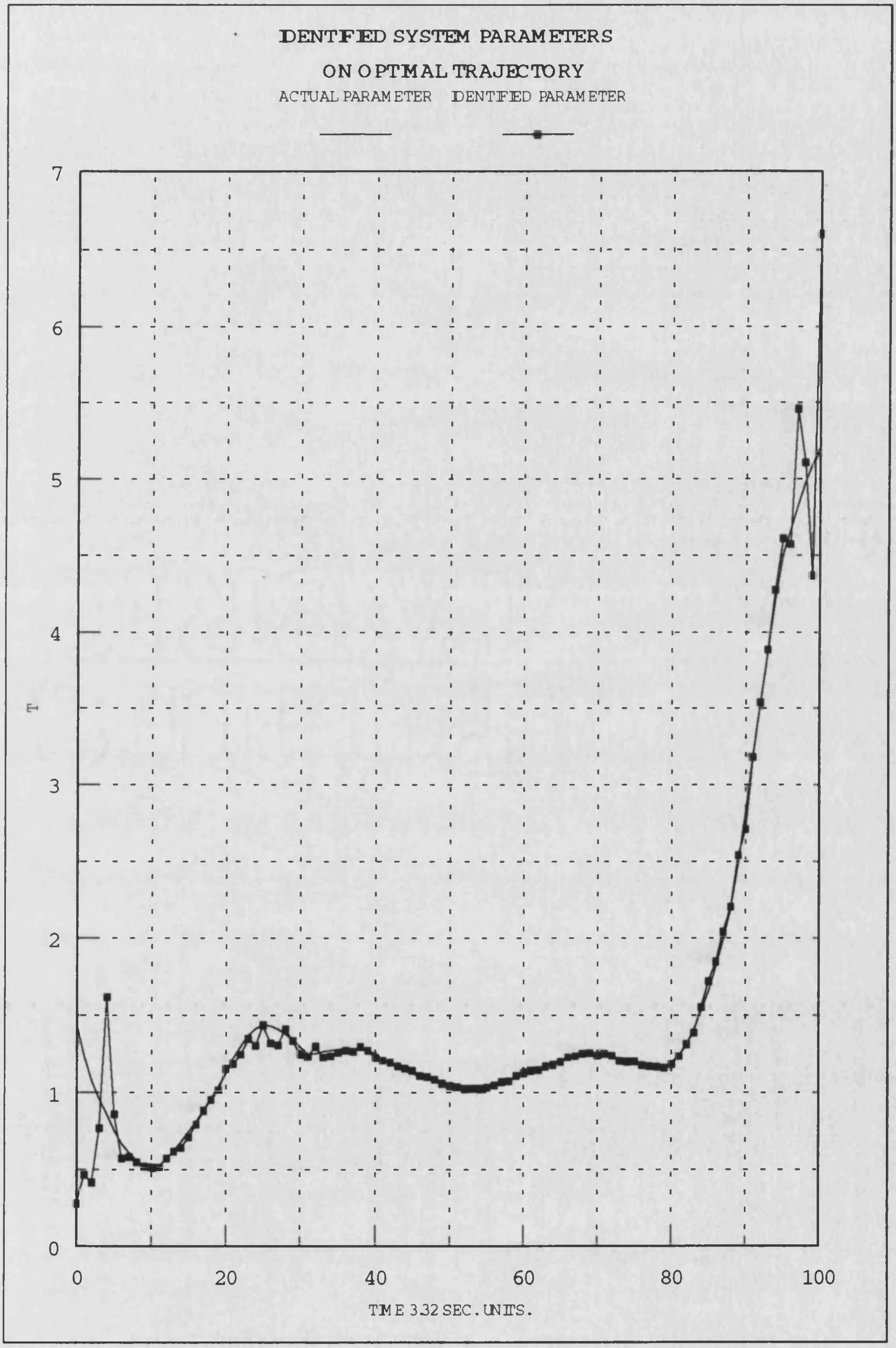

IDENTIFIED SYSTEM PARAMETERS
ON OPTIMAL TRAJECTORY
ACTUAL PARAMETER
IDENTIFIED PARAMETER
7
6
5
4
3
2
1
0
T
0
20
40
60
80
100
TIME 3.32 SEC. UNITS.

Fig. 61

Chapter 8

On-Line Command Stability Augmentation System Adaptation

Chapter 5 has described in detail how the aircraft short-period pitch rate per elevator response varies significantly throughout the optimal climb manoeuvre. In order to reduce the effects of this variation in aircraft response characteristics, a command stability augmentation controller is proposed in this chapter, and the resultant closed-loop aircraft pitch rate per pitch rate demand response investigated.

C.S.A.S Structure:-

A simple proportional plus lagged integral controller with the transfer function as defined below has been considered.

$$\frac{\eta_D(s)}{q_{Error}(s)} = K_c\left[1+\frac{K_i}{s(1+T_c s)}\right]$$

To achieve a uniform closed-loop response characteristic it is necessary to vary the parameters of this controller during the optimal manoeuvre. The difficulty in obtaining a scheduling law for the parameters of the controller as a function of auxiliary state variables, due to the non-linear relationship between aircraft parameters and auxiliary state variables, has already been described. However, by varying the controller parameters as a function of identified aircraft parameters, a simplification of the controller parameter scheduling results. In particular, by selecting the following specific relationships between aircraft identified parameters and controller parameters:-

$$K_c \propto \frac{2\xi}{K_0\omega_n}\text{.,}\quad T_c = \frac{1}{2\xi\omega_n}\text{.,}\quad \text{and}\quad K_i = T_c\omega_n^2$$

the controller parameters will be optimised with respect to the changing dynamics of the aircraft short-period response. The constant of proportionality on $\boldsymbol{K_c}$ defines the design break frequency of the combined aircraft and C.S.A.S. closed-loop response characteristic.

To demonstrate the need to adapt the controller, to compensate for the changing aircraft dynamics, the performance of a fixed parameter C.S.A.S was initially investigated. For this purpose each of the controller parameters was set to the mid-point of their range of optimum values, as defined above and computed from the mid-point of the variation in aircraft parameters during the optimal climb trajectory. The closed-loop step response of the combined fixed parameter C.S.A.S and aircraft has been computed at twenty equally spaced time intervals on the optimal trajectory. The envelope of these step responses is presented in fig.62. Although a significant reduction in the variation of the open-loop response characteristic has been achieved using the fixed parameter C.S.A.S , it is evident that there is still considerable variation in both the natural frequency and damping ratio of the resultant closed-loop pitch rate responses. In fact with the parameters selected for this fixed gain controller the system exhibits instability at some point on the climb trajectory. These instabilities are undoubtedly due to the choice of fixed C.S.A.S integrator gain $\boldsymbol{K_i}$ in conjunction with the value selected for the controller proportional gain $\boldsymbol{K_c}$.

Although not the direct purpose of this thesis, it is certain that this situation could be improved by simply scheduling $\boldsymbol{K_i}$. To this end the optimum variation in $\boldsymbol{K_i}$ is shown against trajectory lapse time together with the optimum Mach number time history in fig. 63. The similarity in shape of these plots of $\boldsymbol{K_i}$ and Mach number indicates that it would be feasible to derive a single valued open-loop gain schedule for $\boldsymbol{K_i}$ as a function of Mach number. This would undoubtedly improve the uniformity of the closed-loop response characteristic on the optimal manoeuvre. This technique however is not exact and open-loop, as has been pointed out, and so the on-line aircraft parameter identification and subsequent C.S.A.S adaptation scheme has been implemented.

Adapted C.S.A.S. Responses:-

The envelope of closed-loop pitch rate step responses obtained using the optimal adaptive C.S.A.S system has been computed at the same twenty time intervals on the optimal climb trajectory. These results are shown in fig. 64 and the close uniformity of all of the step responses is clearly demonstrated. The identification of the aircraft parameters has been achieved using the techniques of Quasilinearisation outlined in chapter 7. The control signal used for the identification of the aircraft parameters was the elevator demand from the C.S.A.S output during the optimal climb manoeuvre and no further test or specific identification signals were required. The C.S.A.S parameters were computed from the identified aircraft parameters, as defined above, and on-line adaptation occurred at the end of each identification interval as a continuous process.

The optimised controller parameter variations as a function of lapse time on the climb manoeuvre are as shown in figs. 65and 66. Also shown in fig. 66 is the variation in the ratio of aircraft lead time constant to C.S.A.S. controller lag time constant during the manoeuvre. Although this ratio varies slightly from unity, indicating that these two time constants are not matched exactly at every point in time on the optimal climb, the deviation from unity is small and hence the subsequent effect on the envelope of closed-loop pitch rate responses is negligible. For the purpose of this investigation the identification time interval was fixed at 200ms. and this proved to be satisfactory. This identification interval was based on the selected closed-loop break frequency which was set arbitrarily at 4 rad. per sec. It should be noted that it is perfectly feasible to vary the identification interval, perhaps as a function of the identified natural frequency of the aircraft; however this additional versatility proved unnecessary in the investigation. The dynamics of the elevator power control were omitted from this investigation purely for simplification purposes. This however has no effect on the validity of the techniques demonstrated. Chapter 9 investigates an additional augmented control scheme suitable for returning an adapted system response to a nominal transient response in an optimum manner.

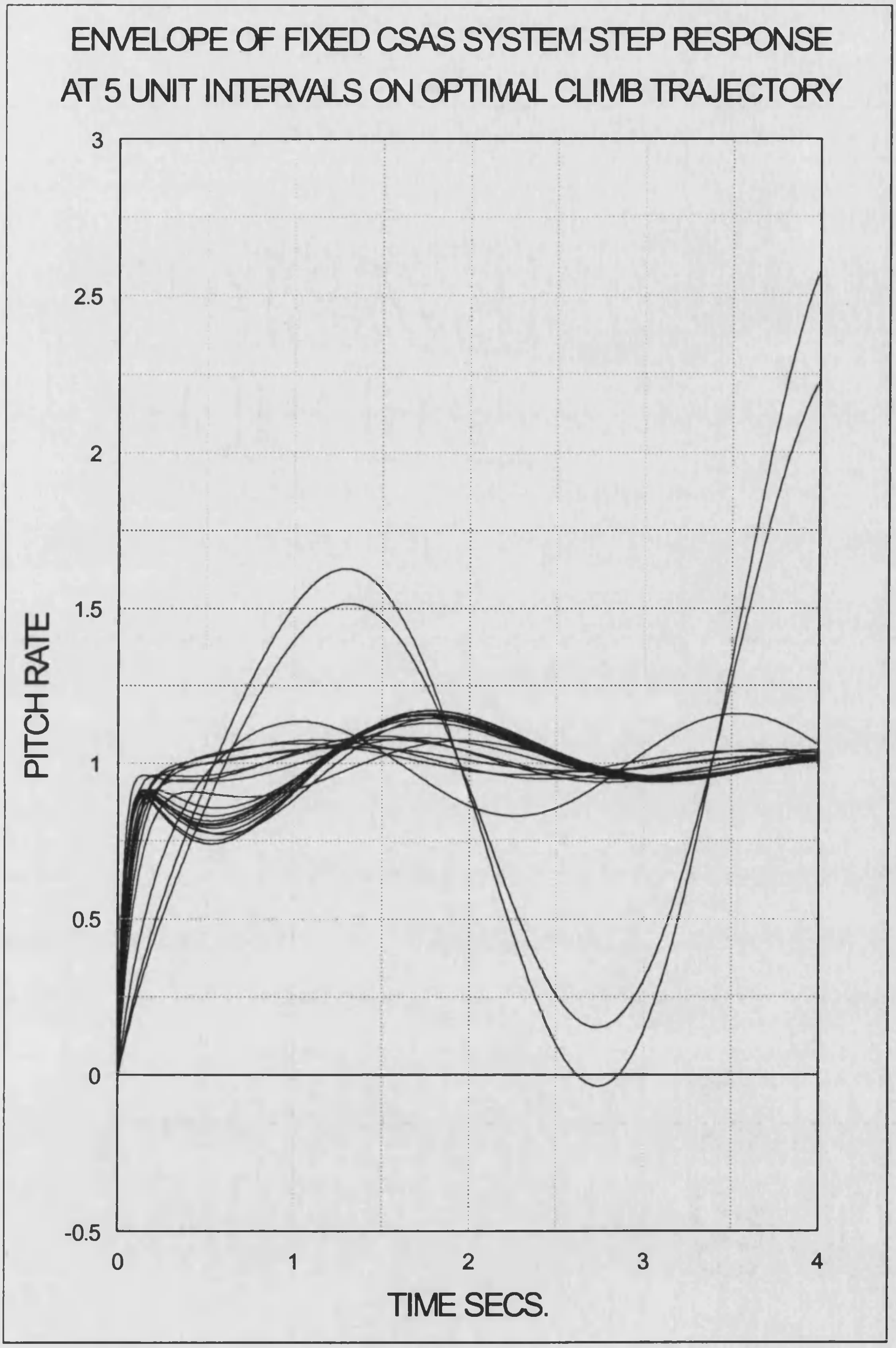
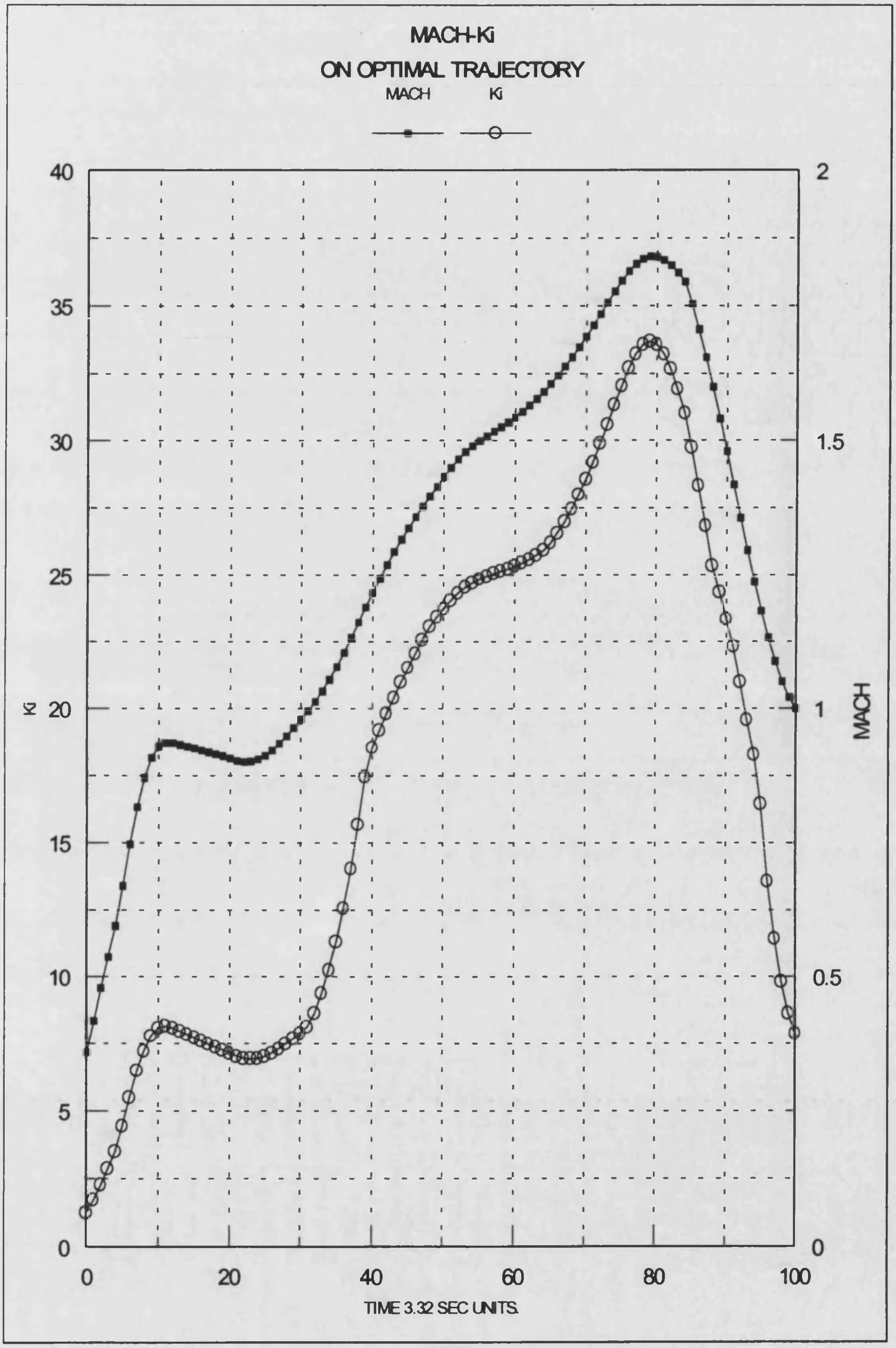
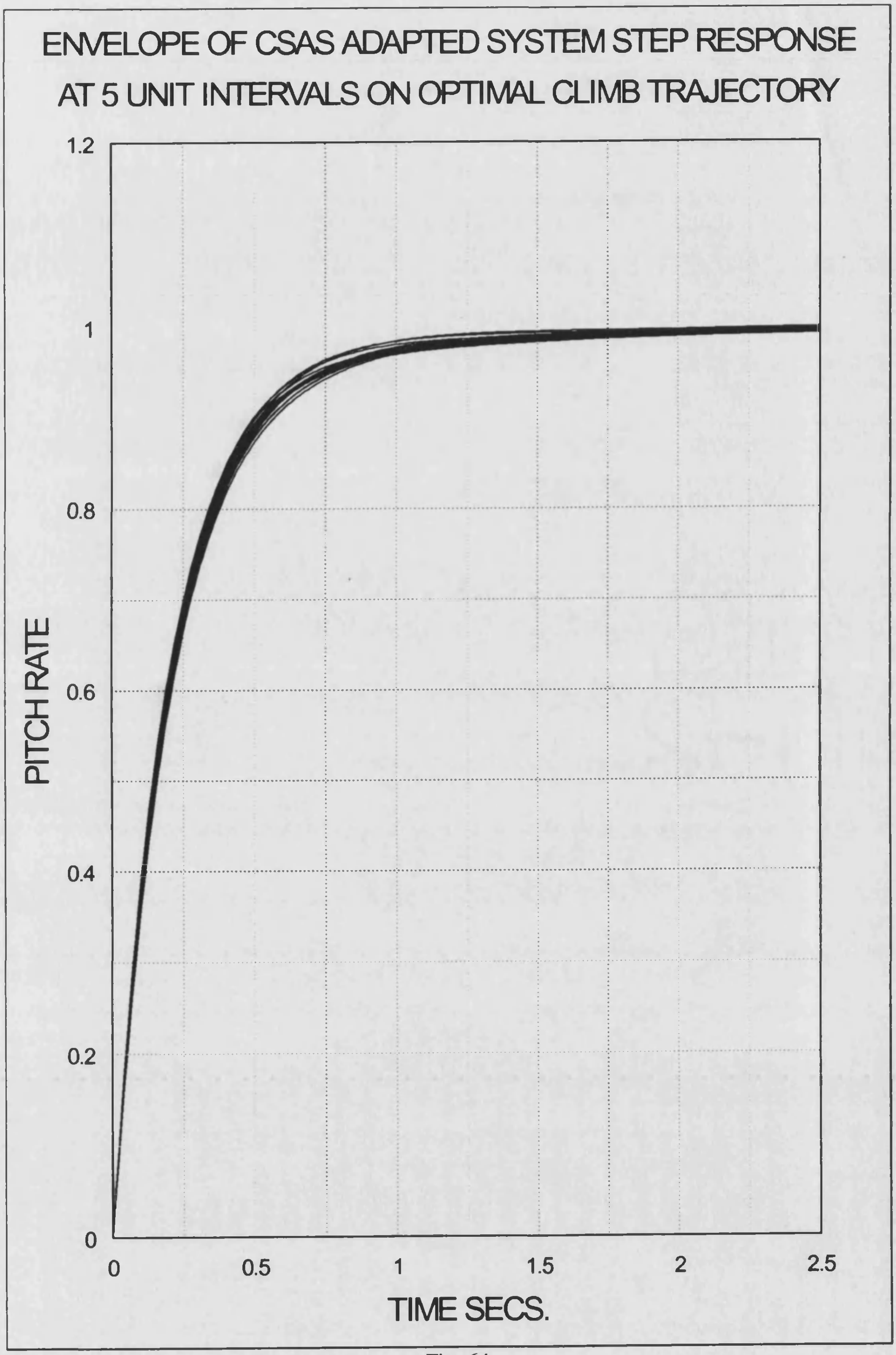

ENVELOPE OF FIXED CSAS SYSTEM STEP RESPONSE
AT 5 UNIT INTERVALS ON OPTIMAL CLIMB TRAJECTORY
PITCH RATE
3
2.5
2
1.5
1
0.5
0
-0.5
0
1
2
3
4
TIME SECS.

Fig. 62



MACH-Ki
ON OPTIMAL TRAJECTORY
MACH
Ki
Ki
MACH
TIME 3.32 SEC UNITS.
0
5
10
15
20
25
30
35
40
0
0.5
1
1.5
2
0
20
40
60
80
100

Fig. 63



ENVELOPE OF CSAS ADAPTED SYSTEM STEP RESPONSE
AT 5 UNIT INTERVALS ON OPTIMAL CLIMB TRAJECTORY
PITCH RATE
TIME SECS.
1.2
1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
2.5

Fig. 64

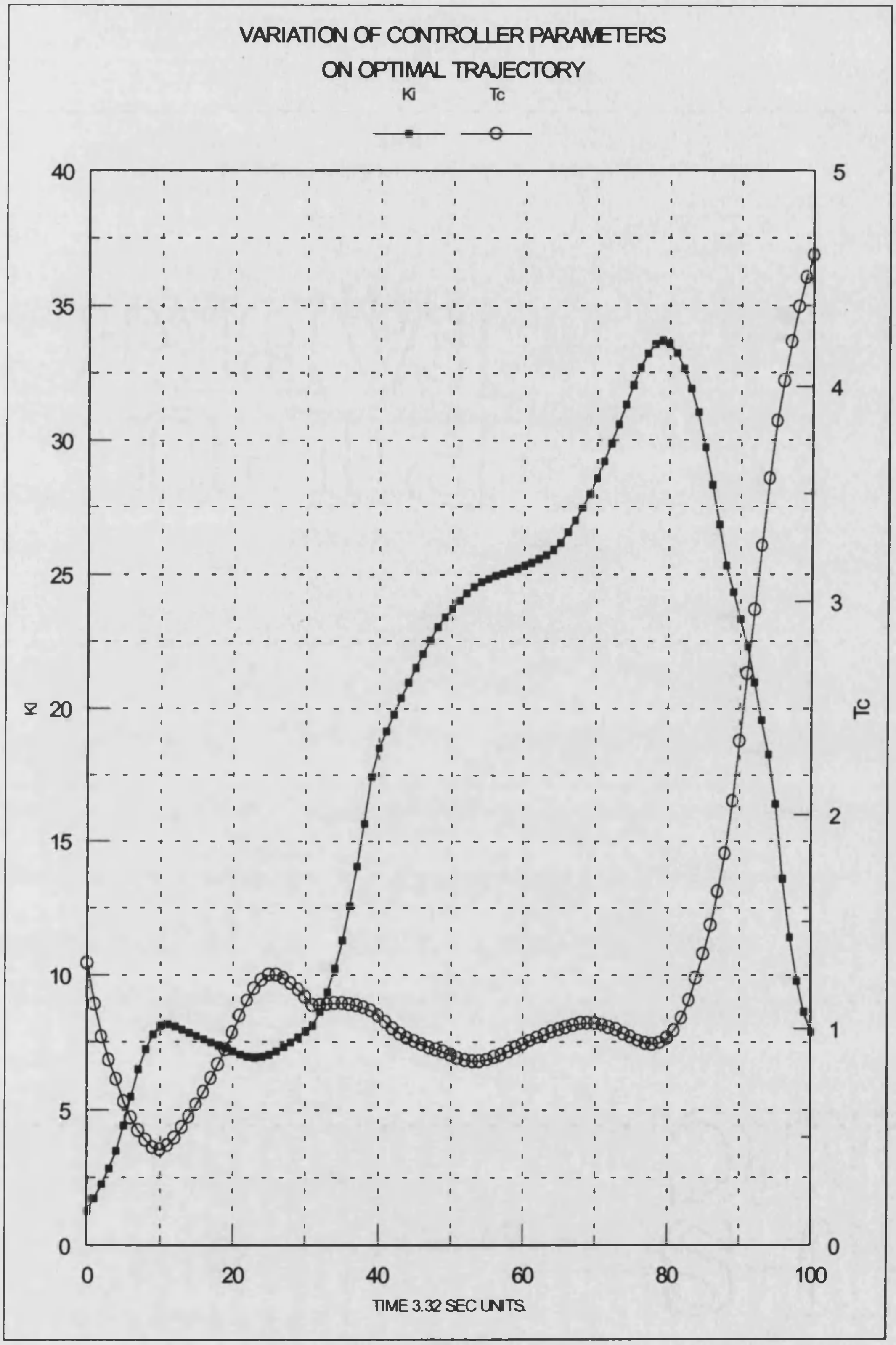
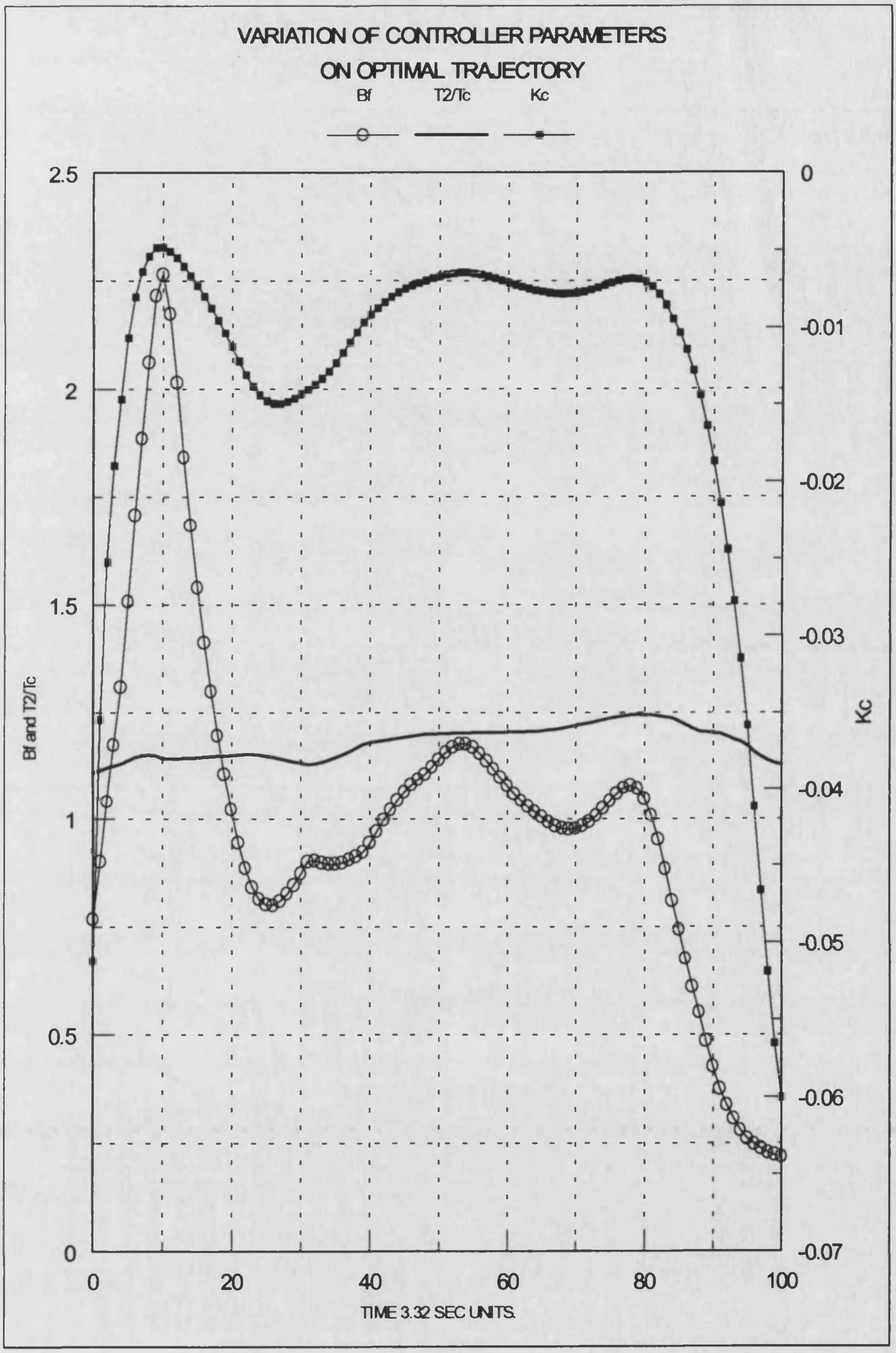

VARIATION OF CONTROLLER PARAMETERS
ON OPTIMAL TRAJECTORY
Ki
Tc
Ki
Tc
TIME 3.32 SEC UNITS.
0
5
10
15
20
25
30
35
40
0
1
2
3
4
5
0
20
40
60
80
100

Fig. 65



VARIATION OF CONTROLLER PARAMETERS
ON OPTIMAL TRAJECTORY
Bf
T2/Tc
Kc
2.5
2
1.5
1
0.5
0
0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
-0.07
0
20
40
60
80
100
Bf and T2/Tc
Kc
TIME 3.32 SEC UNITS.

Fig. 66

Chapter 9

Minimisation Of Adapted Transient Response Errors From A Nominal By Optimal Augmented Feedback Control.

The thesis so far has presented results of investigations on the design, implementation and performance of an on-line adaptive control command stability augmentation system. The overall objective of this work has been to provide a uniform response characteristic for the pitch short period mode control of an aircraft on an optimal climb trajectory. The adaptive process utilises an on-line identification scheme to identify and track the changing aircraft parameters. The identification and adaptation are continuous and are both performed well within the closed-loop transient response time of the system.

At the end of each identification and adaptation interval the closed-loop dynamics, of the combined controller and aircraft combination, match a predetermined nominal system dynamics. The response of the system to subsequent inputs will therefore be as required and also aircraft handling qualities will be as desired. However since the identification and adaptation process requires a finite time before the aircraft parameters are determined, the actual time response of the closed-loop system can significantly deviate from a prescribed nominal transient response. This is due to the possible mismatch between controller and aircraft parameters during the identification interval. An example of this situation is demonstrated in fig. 67. In this figure the response of each of two such initially mismatched controllers is shown. In one example the controller parameters are a factor of ten too large, while in the other example they are a factor of ten too small. This represents a total gain variation of one hundred to one. In each case, at the end of the short identification interval, the controller parameters have been adapted to their nominal values. From this point onwards in the time responses the combined controller aircraft dynamics are correct. It is clearly seen however that the actually achieved time responses differ significantly from the desired nominal transient response. The question then arises whether to accept this situation since the response to subsequent inputs after adaptation will be satisfactory, or whether to attempt to force the system response back onto the nominal transient response trajectory. Since the adaptation process is operating well within the transient response time of the system it would seem desirable to utilise this fact and to restore the actual response to the nominal in some optimum fashion. Since, from the instant of identification and adaptation, the dynamics of the closed-loop system are completely defined, it is therefore possible to design an additional optimal feedback loop. This augmented control can be designed to minimise a quadratic function of the error between the actual system response and the desired nominal transient response.

This additional optimal controller has been implemented and the results achieved are shown in fig. 68. In this figure it is clearly demonstrated that in both initial mismatch configurations the system responses are returned to the nominal transient response characteristic shortly after the identification interval. In these examples the augmented optimal control is only active from the end of the identification interval, and after the complete system dynamics have been determined.

It is evident that the system response accelerations may be unacceptable. These however could be controlled by incorporating additional state constraints in the optimisation performance index.

It has been found beneficial to have this augmented optimal controller active continuously, including during the identification interval. The results obtained in this case are presented in fig. 69 and in expanded view in fig. 70. It can be seen clearly that the deviations from the nominal transient response trajectory are significantly reduced, even though there is a mismatch of controller and aircraft dynamics during the identification interval.

A schematic diagram representing the complete on-line identification process, the adaptive controller, and the optimal augmented controller to minimise errors between system response and a desired nominal is given in fig. 71.

The conclusions drawn from this research together with suggestions for future research are discussed in chapter 10.

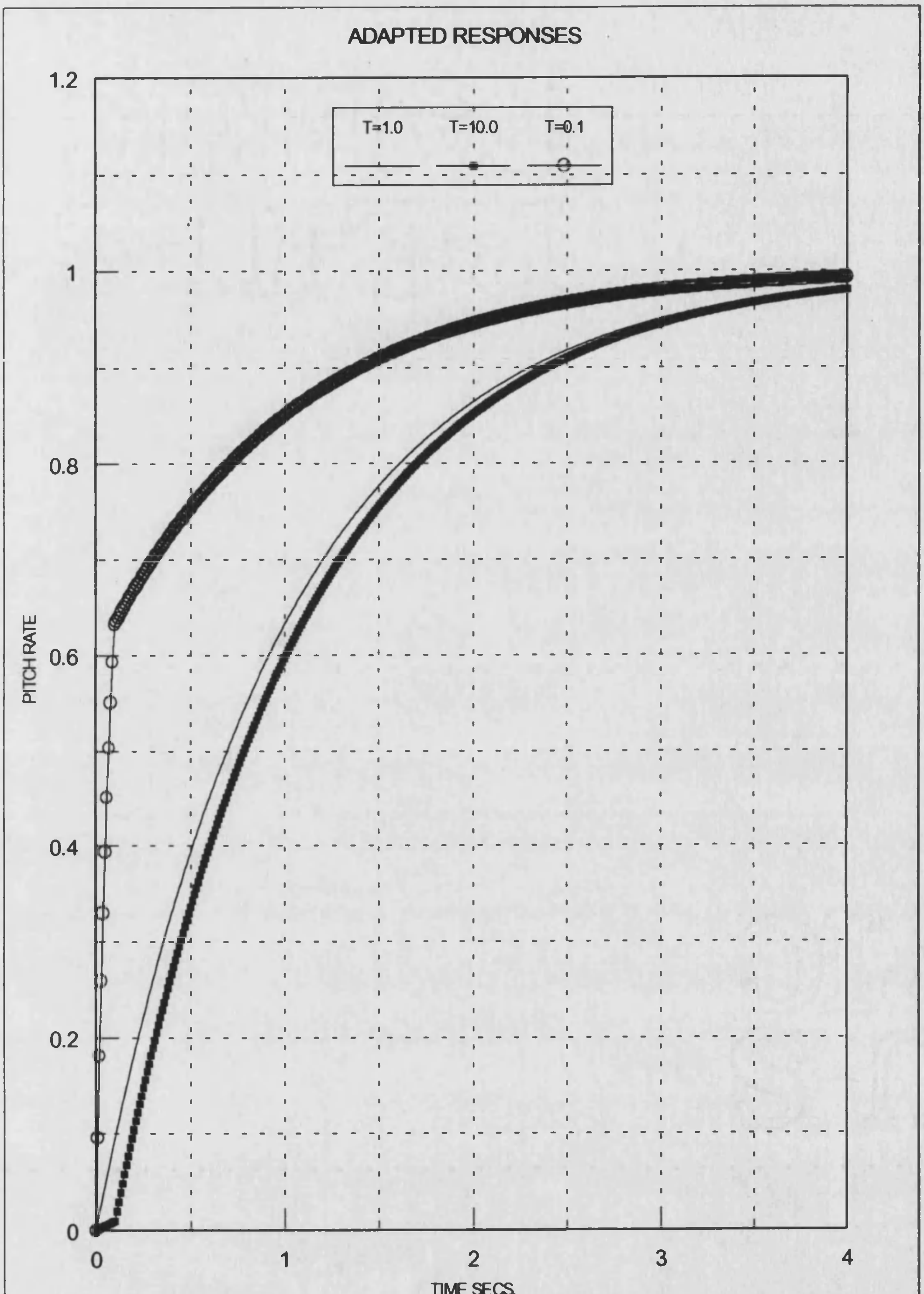
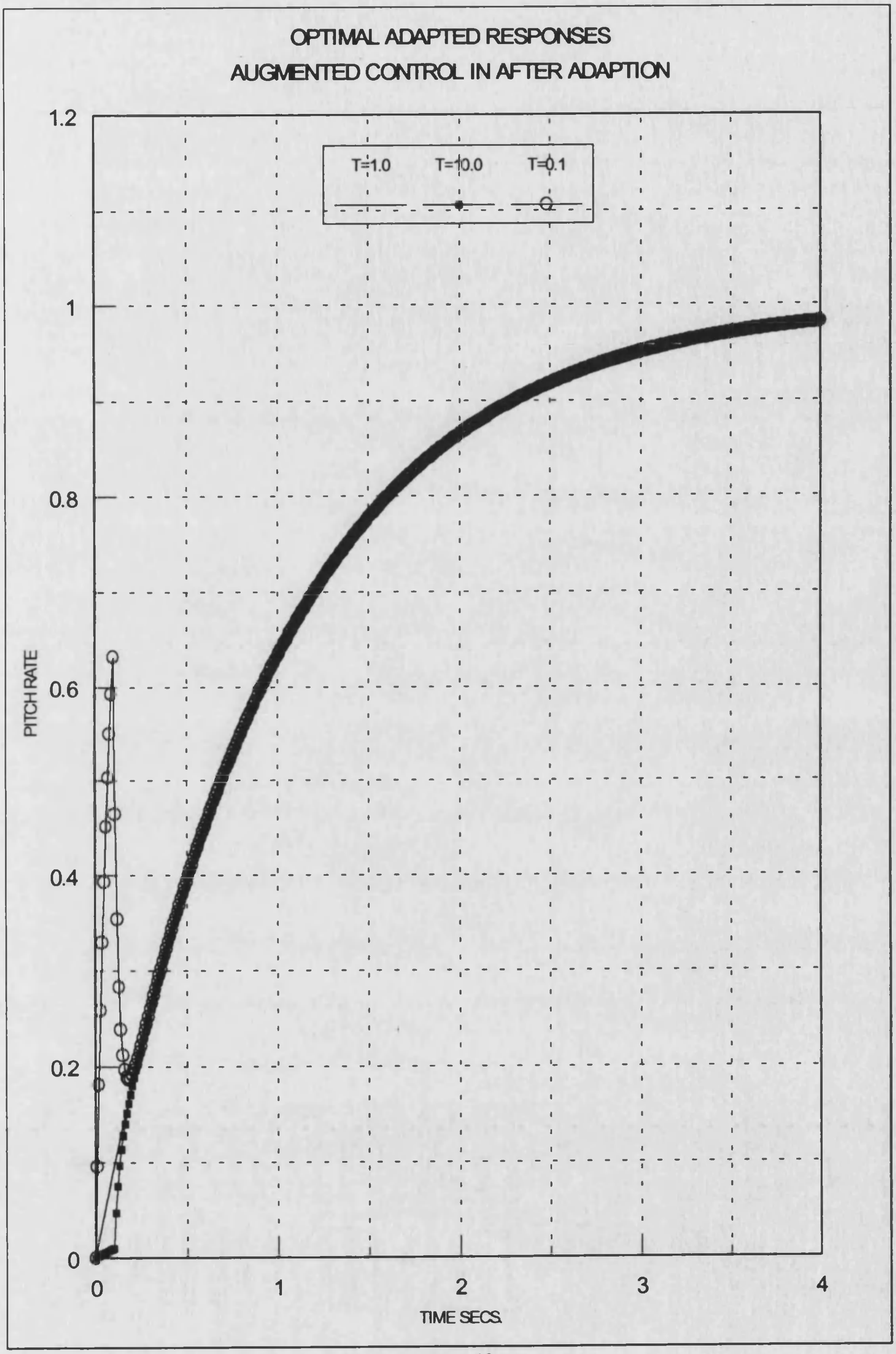

ADAPTED RESPONSES
T=1.0
T=10.0
T=0.1
1.2
1
0.8
0.6
0.4
0.2
0
PITCH RATE
0
1
2
3
4
TIME SECS.

Fig. 67



OPTIMAL ADAPTED RESPONSES
AUGMENTED CONTROL IN AFTER ADAPTION
T=1.0
T=10.0
T=0.1
1.2
1
0.8
0.6
0.4
0.2
0
PITCH RATE
0
1
2
3
4
TIME SECS.

Fig. 68

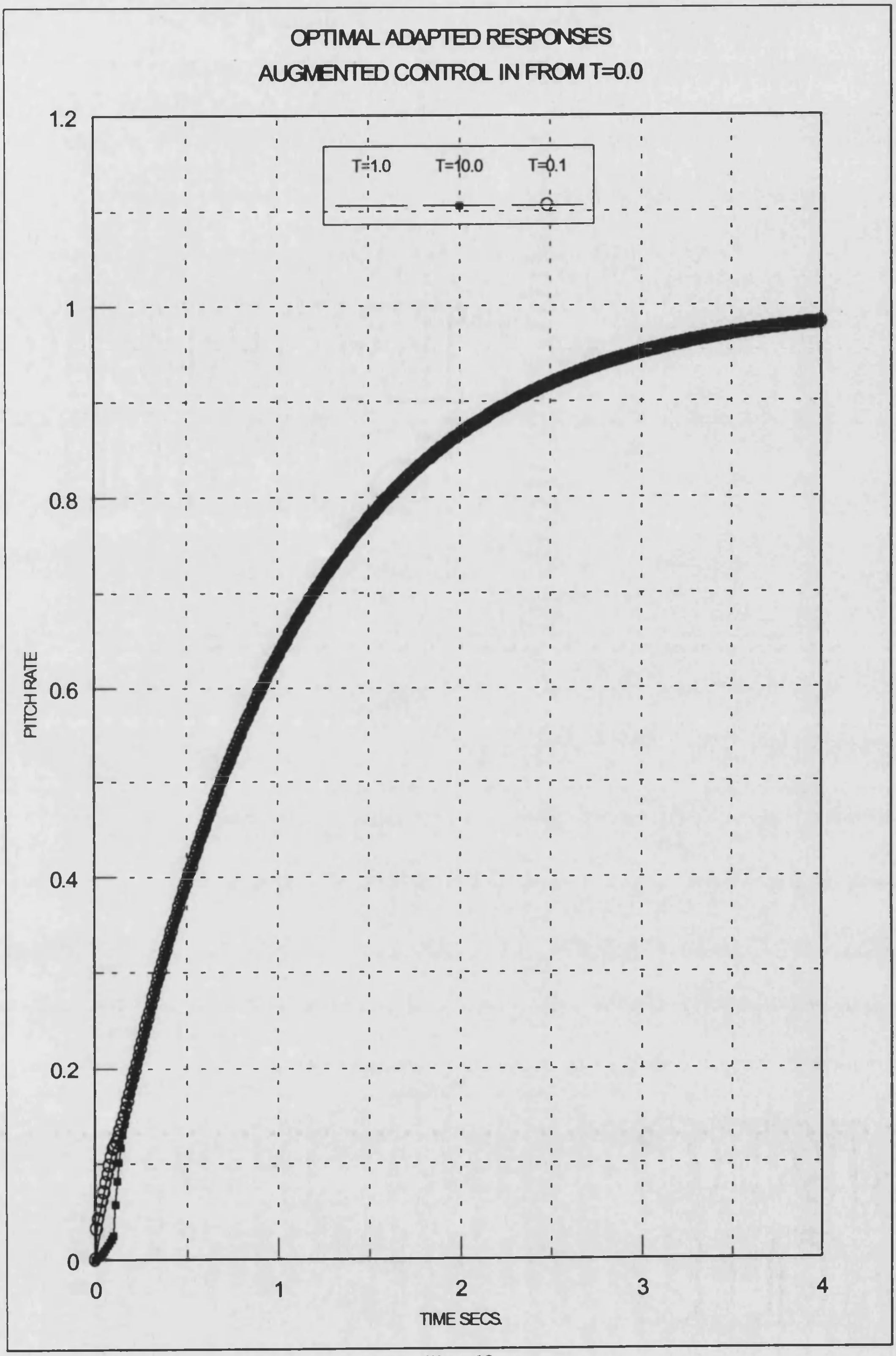

OPTIMAL ADAPTED RESPONSES
AUGMENTED CONTROL IN FROM T=0.0
T=1.0
T=10.0
T=0.1
PITCH RATE
TIME SECS.
1.2
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4

Fig. 69

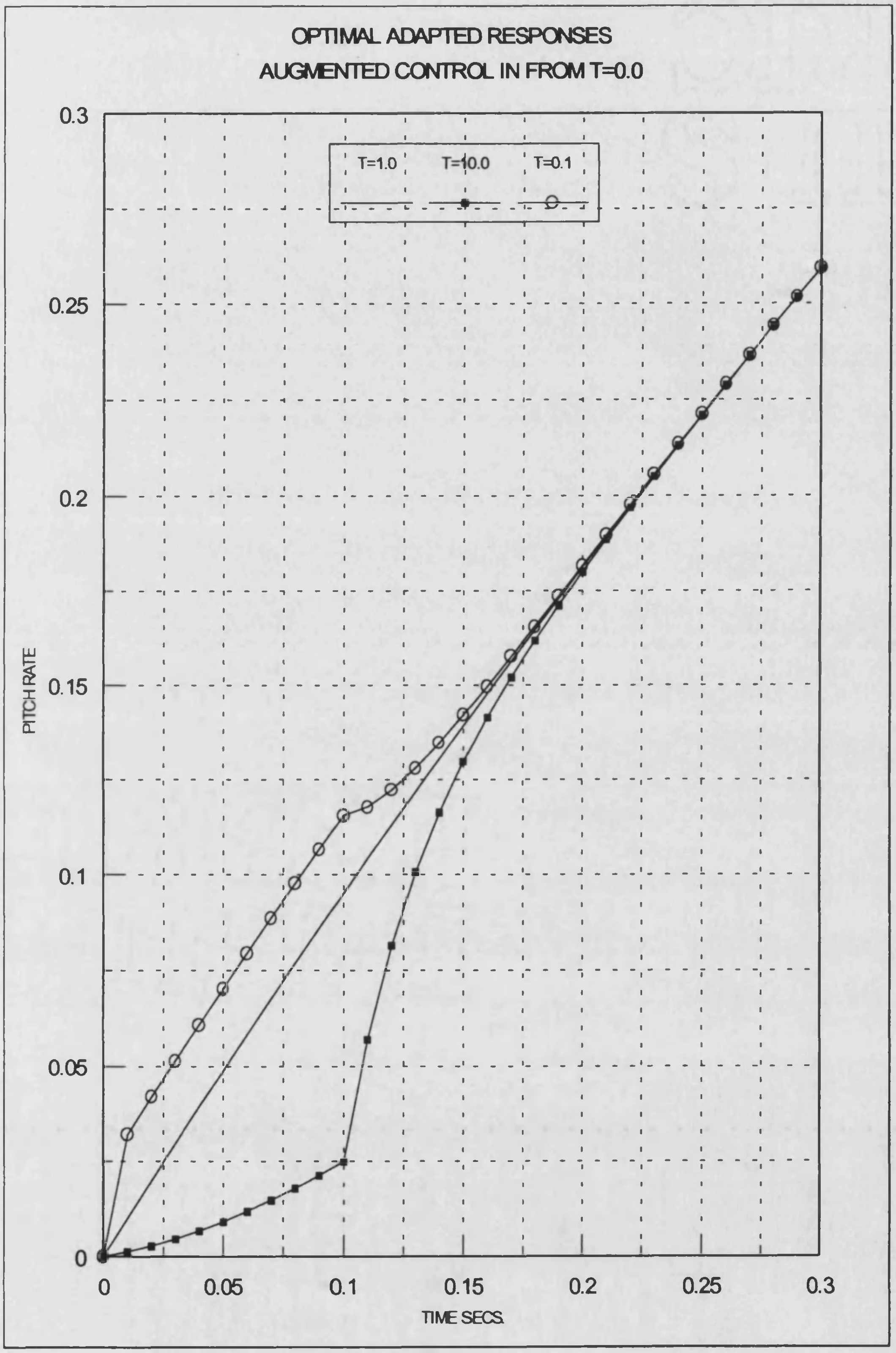
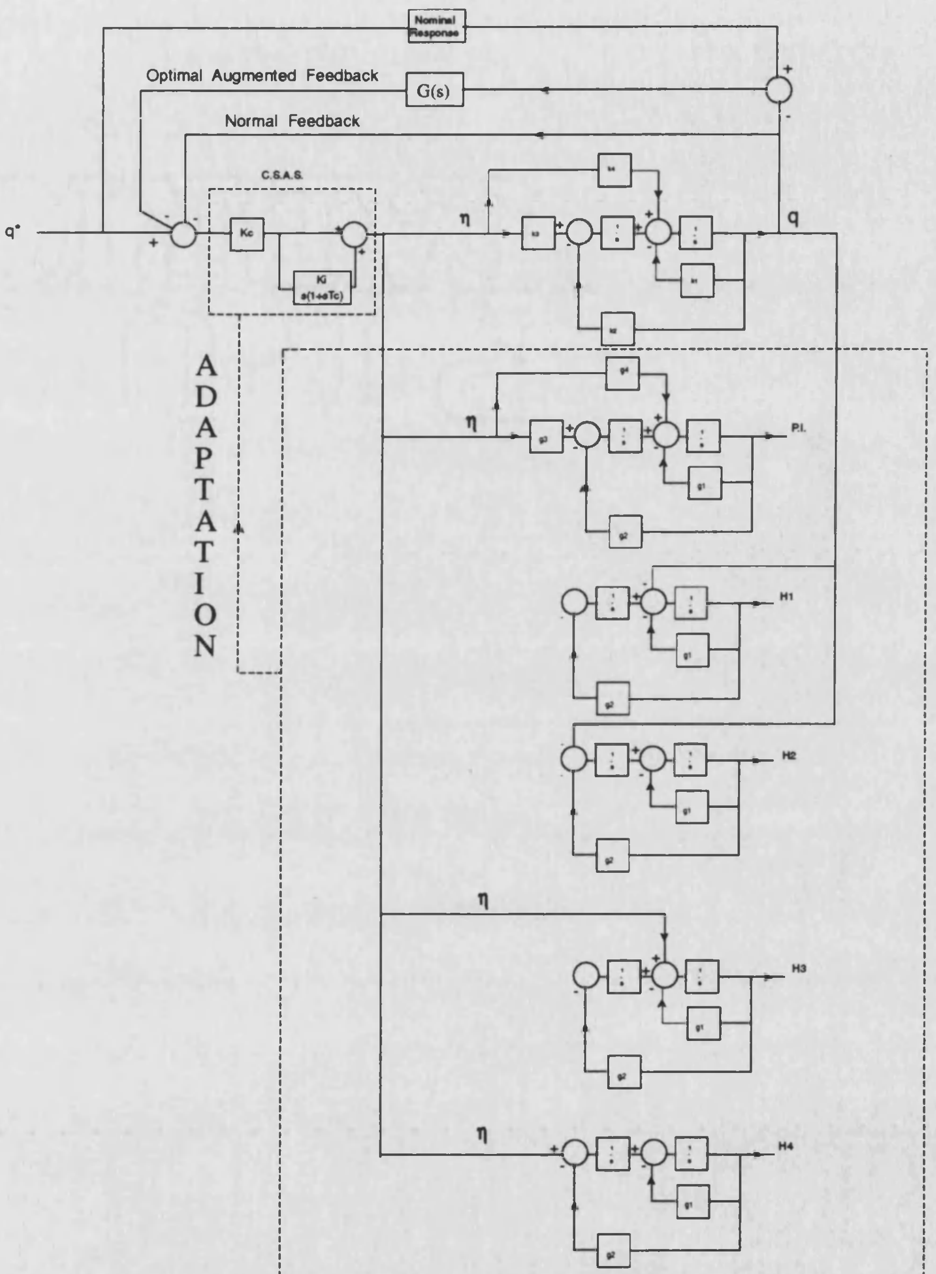

OPTIMAL ADAPTED RESPONSES
AUGMENTED CONTROL IN FROM T=0.0
T=1.0
T=10.0
T=0.1
PITCH RATE
TIME SECS.
0
0.05
0.1
0.15
0.2
0.25
0.3

Fig. 70



On Line Optimal Adaptive Pitch Loop Control Scheme

Fig. 71

Chapter 10

Conclusions And Suggestions For Future Research.

This thesis has investigated a possible adaptive flight control scheme which is based on the explicit high speed on-line identification of the parameters of the aircraft dynamics. The identification is performed using normal operating control inputs.
To test the system the specific case of an optimal climb manoeuvre has been investigated. The optimal trajectory obtained as a solution to this problem has been computed using both Steepest Descent and Quasilinearisation numerical methods. This optimal solution takes the aircraft through many rapidly changing flight conditions with subsequent variation in aircraft response characteristics and handling qualities and therefore is a good manoeuvre with which to investigate the effectiveness of both the on-line identification scheme and the adaptive control algorithm.
Some considerable time has been spent by the author in understanding optimal control techniques from first principles and it is therefore appropriate that the theory of this applicable to the specific optimisation problem has been presented. It has also been necessary to generate the numerical computational software and to verify this for both methods of solution which have been investigated and applied to the solution of the optimal control necessary conditions. In particular the Quasilinearisation method of solution of the resultant two point boundary value problem has been applied to the implementation of the on-line identification aircraft parameter identification scheme.
Optimal control theory has also been applied to the task of minimising the quadratic error function of the adapted system response and a nominal desired transient response characteristic.

Some thought has been given to establishing a confidence level in the identified aircraft parameters. This has been achieved by establishing that the matrix of homogeneous solutions in the Quasilinearisation algorithm of the identification process is non-singular. This relates to the persistency of excitation requirement to excite all the modes of the system. If this requirement is not fulfilled no identification is possible, and the above mentioned matrix becomes singular. In the event of this situation arising, the adaptive process would be inhibited until the validity of the identification is again achieved and the adaptation process automatically resumes. It is also possible to establish a validity range for the identified parameters and also to monitor that the rate of change of these parameters is within limits to provide a confidence level in the identification procedure.

Future Research :-

The accuracy of the aircraft parameter identification and tracking has been clearly demonstrated. In this investigation however only the still air performance of the adaptive system has been considered. The effects of both turbulence and measurement noise on the identification process must now be thoroughly investigated. Should the states of the aircraft system be contaminated by noise it may be necessary to introduce a Kalman filtering technique to improve the determination of the system states before the identification algorithm is applied. It is felt that while a time penalty will be introduced to obtain a statistical average or filter the results of the identification, the method demonstrated in this thesis is worthy of further investigation.

Hardware implementation considerations have not been attended to in this thesis. This is because each year seems to bring new developments in data signal processor technology, with both increases in speed and complexity of processing power. At the time of writing the Texas Instrument C40 D.S.P. range of microprocessors seems particularly suited to the on-line identification and adaptation tasks. It is intended to use this processor to further this on-going research activity.

System integrity needs to be developed and a failure analysis of the system performed to establish continuous availability of the system for flight critical conditions.

Much further work needs to be undertaken in the development of this system before it finds universal acceptance. Nevertheless the author believes this research has demonstrated the possible potential of such a closed-loop adaptive system and commends it to the aircraft industry. It is felt that an in-flight demonstrator programme is long overdue in this country, of a controller utilising an on-line identification facility. I would call upon The Defence Research Agency, Avionics and Aircraft manufacturers to pursue this activity which may well prove to enhance future aircraft performance.

References And Bibliography.

1- BLISS G.A., Lectures On The Calculus Of Variations, 1946, The University Of Chicago.

2- BRYSON A.E. and HO YU-CHI, Applied Optimal Control, Optimisation, Estimation and Control; Hemisphere Publishing Corporation 1975.

3- KIRK D.E. Optimal Control Theory; Prentice-Hall 1970.

4- SINHA, N.K. and KUSZTA, B. "On-line Identification of Discrete-Time Systems", in Modelling and Identification of Dynamic Systems. Van Nostrad Reinhold Comp. England (1983).

5- ASTROM, K.J. and WITTENMARK, B., Adaptive Control, Addison Wesley Publishing, U.S.A., 1989.

6- CLARKE, D.W., "Generalized Least-Square Estimation of the Parameters of a Dynamic Model", Preprints of the First IFAC Symposium on Identification. Prague, Czechoslovakia, (1967).

7- HASTING-JAMES, R. and SAGE, M.W. "Recursive Generalized Least-Square Procedure for On-Line Identification of Process Parameters", Proc. IEE 116, pp. 2057-2062, 1969.

8- ASTROM, K.J. and BOHLIN, N.T. "Numerical Identification of Linear Dynamic Systems from Normal Operating Records", Proc. IFAC Symposium on Self-Adaptive Control Systems, 96-110, Teddington, England, 1965.

9- GERTLER J. and BANYASIZ, C. "A Recursive (On-Line) Maximum Likelihood Identification Method", IEEE Trans. Aut. Control, Vol. AC-19, pp. 816-820, 1974.

10- BELLMAN R.E. and KALABA R.E.,Quasilinearisation And Nonlinear Boundary-Value Problems.,Elsevier Publishing Company, Ltd. 1965.

11- YOUNG, P.C. "An Instrumental Variable Method for Real-Time Identification of a Noisy Process", Automatica Vol. 6, pp. 271-287, 1970.

12- YOUNG, P.C. Lectures on the Parameter Estimation, Summer School: Theory and Practice of Systems Modelling and Identification. Toulouse, 1972.

13- BUDIN, M.A. "Minimal Realization of Discrete Linear Systems from Input-Output Observation", IEEE Trans. Aut. Control, Vol. AC-17, pp 252-253 1972

14- SAGE, A.P. Optimum Systems Control, Prentice-Hall, Englewood Cliffs, N.J., 1968.

15- DETCHMENDY, D.M. and SHRIDAR, R. "On the Experimental Determination of the Dynamical Characteristics of Physical Systems", Proc. National Electronics Conf. , pp. 522-530, Chicago, 1965.

16- SARIDIS, G.N., "Comparison of Six On-Line Identification Algorithms", Automatica, Vol. 10, pp. 69-79, 1974.

17- KALABA, R. and SPINGARN, K. "On the Rate of Convergence of the Quasilinearization Method", IEEE Trans. Aut. Control Vol.10, pp. 798-799, 1983.

18- ISERMANN, R., BAUR, U., BAMBERGER, W., KNEPPO, P. and SIEBERT, H. "Comparison of Six On-Line Identification and Parameter Estimation Methods", Automatica, Vol. 10, pp. 81-103, 1974.

19-WHITAKER, H.P. , YAMRON, J. and KEZER, A., Design of Model Reference Adaptive Control System, Report R- 164, 1958, Instrumentation Laboratory, M. I. T., Cambridge. J.K.M.

20- ASTROM, K.J. and WITTENMARK, B. " On Self Tuning Control", Automatica, Vol. 9, 1973, pp. 185-199.

21- ISERMANN, R., "Process Fault Detection Based on Modelling and Estimation Method - A Survey ", Automatica, Vol. 20, No. 4, pp. 387-404, 1984.

22- WONG, K.Y. and POLAK, E. "Identification of Linear Discrete Time Systems Using the Instrumental Variable Methods", IEEE Trans. Aut. Control, Vol. AC-12, 707, 1967.

23- MacCORMAC J.K.M. Reduction in the variation of aircraft response characteristics during optimal trajectory manoeuvres. I.C.A.S. Anaheim 1994. J.K.M.

24- MacCORMAC, J.K.M. A report on research on sampled data adaptive aircraft control systems. University College Swansea 1968 Ministry of Technology (R.A.E.-Farnborough) 1968. J.K.M.

25- MacCORMAC, J.K.M. "The Use of Hybrid Computation in an On-Line Identification Scheme", IFAC, Budapest, 1968. J.K.M.

26- BABISTER, A.W., Aircraft Dynamic Stability and Response, Pergamon Press Ltd. U.K., 1980.

27- BLAKELOCK J.H., Automatic Control of Aircraft and Missiles, John Wiley & Sons, Inc., U.S.A., 1965.

28- ASTROM, K.J. and WITTENMARK, B., Computer Controlled Control Systems", 1984.

29- PARKS, P.C. "Lyapunov Redesign of Model Reference Adaptive Control Systems", IEEE Trans. Auto. Cont., Vol. AC-11, No. 3, 1966, pp.362-367.

30- SODERSTROM T. and STOICA P., System Identification. Prentice-Hall 1989.

31- MONOPOLI, R.V. "Model Reference Adaptive Control with an Augment Error Signal", IEEE Trans. Auto. Cont., 1974, pp. 474-483.

32- ELLIOT, H. "Direct Adaptive Pole Placement with Application to Nonminimum Phase Systems", IEEE Trans. Auto. Cont., 1982, pp. 720-722.

33- CLARKE, D.W. and GAWTHROP, P.J. "Self Tuning Controller", IEE Proc., 1975, Vol. 122, No. 9, pp. 929-934.

34- CLARKE, D.W. "Self Tuning Control of Nonminimum Phase System", Automatica, Vol. 20, 1985, No. 5, pp. 501-517.

35- CLARKE, D.W., KANJILAL, P.D. and MOHTADI, C. "A Generalized LQG Approach to Self-Tuning Control", Part 1., Int. J. Control, 1985, Vol. 41, No. 6, pp. 1509-1523.

36- ASTROM, K.J. and WITTENMARK, B. "LQG Self Tuner", IFAC Symposium on Adaptive Control, 1983, San Fransisco, U.S.A..

37- GRIMBLE, M.J., "Controllers for LQG Self-Tuning Applications with Coloured Measurement Noise and Dynamic Costing', IEE Proc., 1986, Vol. 133, Pt. D, No. 1, pp. 19-29.

38- HUNT, K.J., GRIMBLE, M.J. and JONES, R.W., LQG Feedback and Feedforward Self-Tuning Control ,IFAC 10th Triennial World Congress, Munich, FRG, 1987, pp. 169-174.

39- GOODWIN, G.C. and MAYNE, D.Q., "A Parameter Estimation Perspective of Continuous Time Model Reference Adaptive Control", Automatica, Vol. 23, 1987, pp. 57-70. X

40- ROSENBERG K., Private Communication, 1991. (Turbulence Ref) X for Murat.

41- ETKIN, B., Dynamics of Atmospheric Flight, John Wiley & Sons, Inc., 1972. X

42- Third-Generation TMS320 User's Guide, Texas Instruments Corp., 1988. JLM, (X)

43- GERALD, C.F. and WHEATLEY, P.O., Applied Numeric Analysis, Addison-Wesley Publishing Comp., 1989. X

44-SASTRY. S. and BODSON. M., Adaptive Control-Stability, Convergence and Robustness, Prentice-Hall International. 1989. X

45-NARENDRA.K.S. and ANNASWAMY. A.M.,Stable Adaptive Systems, Prentice-Hall International.1989. X

46-BELL.D. and GRIFFIN A.W.J., Modern Control Theory And Computing. McGraw-Hill 1969. JLM (X)

47-BREAKWELL J.V., SPEYER J.E.L., and BRYSON A.E., Optimisation and Control of Non-linear Systems Using the Second Variation. J.S.I.A.M. Control 1963.

48-GOODWIN G.C. and SIN K.S., Adaptive Filtering Prediction and Control. Prentice-Hall 1984.

49- McLEAN D., Automatic Flight Control Systems. Prentice-Hall 1990.

50- BREAKWELL J.V. The optimisation of trajectories. J.S.I.A.M. 1959

51-McGILL R. Optimal control, inequality constraints, and the generalised Newton-Raphson algorithm. J.S.I.A.M. 1965.

52- BRYSON A.E., DESAI M.N. and HOFFMAN. Energy state approximation in performance optimisation of supersonic aircraft. J. Aircraft 6: 181-8. 1969

53- McGILL R. and KENNETH P., Solution of variational problems by means of the generalised Newton-Raphson operator. A.I.A.A. Journal vol 2., 1964

54- KUMAR K.S. and SRIDHAR R., On the identification of control systems by the Quasilinearisation method. I.E.E.E. Trans. Automatic Control 152-4; 1964.

Appendix A

Research Related Publications

by

J.K.M. MacCORMAC

REDUCTION IN THE VARIATION OF AIRCRAFT RESPONSE CHARACTERISTICS DURING OPTIMAL TRAJECTORY MANOEUVRES

J.K.M. MacCormac

School of Electrical and Electronic Engineering
University of Bath
Bath England BA2 7AY

Abstract:-Aircraft trajectory optimisation frequently results in significant variation in the short period response of the aircraft throughout the optimal manoeuvre. The optimisation problem of acquiring maximum height in a fixed time while minimising a function of drag and satisfying a desired terminal constraint on velocity is considered . The resultant two-point boundary value problem is solved by a combination of the methods of steepest descent and quasilinearisation. The variation of the short-period dynamics on the optimal trajectory is investigated and in this example the steady-state gain, damping ratio, natural frequency and lead time constant vary by factors of up to ten to one. Scheduling of a command stability augmentation system with respect to auxiliary variables such as dynamic pressure, mach number and height, reduces this variation . It is shown that on the optimal trajectory the gain scheduling is not single- valued, resulting in a complex non-linear gain adjustment algorithm. A unique relationship between aircraft parameter variations and controller gains is determined and the combination of these provides a uniform pitch rate response characteristic throughout the optimal trajectory. The paper investigates the use of a quasilinearisation based algorithm for the on-line identification and tracking of the aircraft parameters. The subsequent adaptation and re-optimisation of the controller is performed to minimise the error between a desired optimal transient pitch rate response and the actual system response. This re-optimisation of system performance is achieved using an on-board digital model of the identified aircraft.

Introduction

This paper investigates the variation of aircraft response characteristics during an optimal trajectory manoeuvre. The optimal climb manoeuvre of maximising height acquired in a fixed time while satisfying a desired terminal constraint on the final velocity and minimising a function of drag has been chosen as the starting point for this investigation. This optimal manoeuvre has been specifically chosen as the aircraft encounters a significant portion of the flight envelope in performing the task. Also the solution of a similar optimisation problem is available[1,2] and it has therefore been possible to verify optimisation software developed for use in this investigation. In particular the variation of the aircraft parameters defining the small perturbations equations of motion representing the short period pitch response of the aircraft are investigated throughout the optimal trajectory manoeuvre. It is shown that the aircraft parameters are not in general single valued with respect to auxiliary variables such as dynamic pressure, mach number etc.; hence it is difficult to determine a satisfactory gain scheduling control law for a command stability augmentation system which will provide a uniform response characteristic throughout the manoeuvre. An on-line identification scheme is investigated to identify and track the parameters during the manoeuvre and it is these identified parameter values which are used to adapt the C.S.A.S. parameters in place of the normal auxiliary variables.
A relationship between aircraft and controller parameters is obtained which significantly reduces the variation in aircraft closed-loop pitch rate to pitch rate demand response throughout the trajectory. This closed loop adaptive system operates within the transient response time of the aircraft and maintains the transient response uniform for subsequent command inputs. It should be noted however that a finite identification period is required to establish the aircraft parameters and adapt the C.S.A.S. During this period the system transient response can deviate from the desired nominal transient response. At the end of the identification interval the adapted system dynamics are defined and this information is used to augment the control to correct these deviations in the response and return the transient response to the nominal desired transient response in an optimised manner. The augmented control which minimises an error function between desired nominal and actual transient response can also be operative during the identification and adaptation interval. This optimal adaptive controller reduces the deviations in the initial transient response from the nominal when there may be substantial mismatch between the dynamics of

the closed loop adaptive system and those of the nominal transient response.

Climb Optimisation

The necessary conditions for optimal control to minimise a generalised cost functional of the form

$$J=\phi(\underline{x}(t_f),t_f)+\underline{\gamma}^T\psi(\underline{x}(t_f),t_f)+\int_{t_0}^{t_f}\{L(\underline{x},\underline{u},t)+\underline{\lambda}^T(f(\underline{x},\underline{u},t)-\dot{\underline{x}})\}dt$$

subject to dynamic constraints $\dot{\underline{x}}(t) = f(\underline{x},\underline{u},t)$ and specified terminal conditions are given below.

The Euler Lagrange equations:

$$\dot{\underline{\lambda}} = -(\frac{\partial f}{\partial x})^T \underline{\lambda} - (\frac{\partial L}{\partial x})^T$$

The optimality condition for unconstrained controls

$$\frac{\partial L}{\partial u} + \underline{\lambda}^T(\frac{\partial f}{\partial u}) = 0$$

The co-state terminal conditions

$$\underline{\lambda}^T(t_f) = (\frac{\partial \phi}{\partial x})_{t_f} + \underline{\gamma}^T(\frac{\partial \psi}{\partial x})_{t_f}$$

A fixed time problem has been considered in this instance for ease of computation. The actual cost functional chosen to be minimised for this maximum height in a fixed time problem was chosen as

$$J = -h(t_f) + \gamma_1(V(t_f) - 968.) + \frac{1}{2}\int_{t_0}^{t_f} \alpha^2 dt.$$

The initial conditions used for the state equations were as defined in the boundary conditions for the study. The optimisation period used was 332.0 sec. The system states were unconstrained to simplify the problem. The variation in engine thrust characteristics and aerodynamic data with Mach number are as shown in (Figs 16-17) and an interpolation procedure was used to generate the appropriate values and required partial derivatives at each time step in the integration process. From the aircraft forces diagram (Fig. 1) and applying the above necessary conditions the state and co-state equations are as follows.

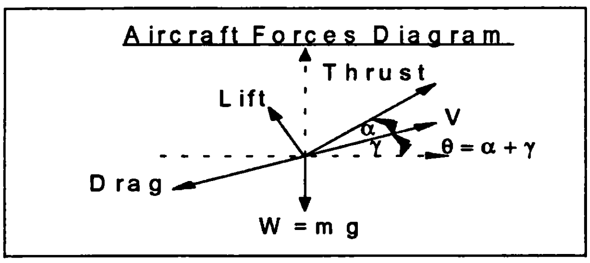


FIGURE 1

The State and Co-state Equations

$$\dot{V}=\frac{1}{m}\{T\cos\alpha - D - mg\sin\gamma\}$$

$$\dot{\gamma}=\frac{1}{mV}\{L+T\sin\alpha - mg\cos\gamma\}$$

$$\dot{h}=V\sin\gamma$$

$$\dot{x}=V\cos\gamma$$

$$\dot{m}=-\frac{T}{cg}$$

$$\dot{\lambda}_V=\frac{\lambda_V}{m}(\frac{\partial D}{\partial V}-\frac{\partial T}{\partial V}\cos\alpha)+\frac{\lambda_\gamma}{mV^2}\{(L+T\sin\alpha-mg\cos\gamma) \\ -V(\frac{\partial T}{\partial V}\sin\alpha+\frac{\partial L}{\partial V})\}-\lambda_h\sin\gamma-\lambda_x\cos\gamma+\frac{\lambda_m}{cg}\frac{\partial T}{\partial V}$$

$$\lambda_\gamma=\lambda_V g\cos\gamma-\lambda_\gamma\frac{g}{V}\sin\gamma-\lambda_h V\cos\gamma+\lambda_x V\sin\gamma$$

$$\dot{\lambda}_h=\frac{\lambda_V}{m}(\frac{\partial D}{\partial h}-\frac{\partial T}{\partial h}\cos\alpha)-\frac{\lambda_\gamma}{mV}(\frac{\partial T}{\partial h}\sin\alpha+\frac{\partial L}{\partial h})+\frac{\lambda_m}{cg}\frac{\partial T}{\partial h}$$

$$\dot{\lambda}_x=0$$

$$\dot{\lambda}_m=\frac{\lambda_V}{m^2}(T\cos\alpha-D)+\frac{\lambda_\gamma}{m^2V}(L+T\sin\alpha)$$

The Optimal Control

The optimal control obtained from the optimality condition is given by

$$\alpha - \frac{\lambda_V}{m}\{T\sin\alpha+\frac{\partial D}{\partial\alpha}\}+\frac{\lambda_\gamma}{mV}\{\frac{\partial L}{\partial\alpha}+T\cos\alpha\}=0$$

The Boundary Conditions

$V(t_0)=400.0\,ft.sec^{-1}$	$\lambda_V(t_f)=\gamma_1$
$\gamma(t_0)=0.0\,rad.$	$\lambda_\gamma(t_f)=0.0$
$h(t_0)=700.0\,ft.$	$\lambda_h(t_f)=-1.$
$x(t_0)=0.0\,ft.$	$\lambda_x(t_f)=0.0$
$m(t_0)=1304\,.slugs$	$\lambda_m(t_f)=0.0$

This complete set of equations constitute a non-linear two-point boundary value problem. The solution has been obtained by a combination of both steepest-descent and quasilinearisation iterative computational techniques. In the steepest-descent method a starting vector was chosen for the control and the state equations were integrated forward in time. At the end of the optimisation interval the terminal condition on $\lambda_V(t_f)$ was set to a weighted function of the error between the computed and desired terminal value of $V(t_f)$ and the co-state equations were integrated backwards in time using the solution of the state equations obtained in the forward integration. A new control vector was computed from

$$\alpha_{N+1} = \alpha_N - \tau\left[\alpha - \frac{\lambda_V}{m}\{T\sin\alpha + \frac{\partial D}{\partial\alpha}\} + \frac{\lambda_\gamma}{mV}\{\frac{\partial L}{\partial\alpha} + T\cos\alpha\}\right]_N$$

where τ controlled the step length along the gradient, and the process was repeated until the terminal error on $V(t_f)$ was within a small norm. To test the quasilinearisation programme which is required for the on-line identification process, the state and co-state equations were first linearised about the solution obtained from the steepest descent procedure which was then used as a starting vector for the quasilinearisation method of solution of the two-point boundary value problem. This process was iterated to convergence and a small improvement was obtained in the maximum acquired height with the terminal condition on $V(t_f)$ exactly satisfied. The results obtained by both methods of the optimum height versus Mach profiles are shown in (Fig 2.)

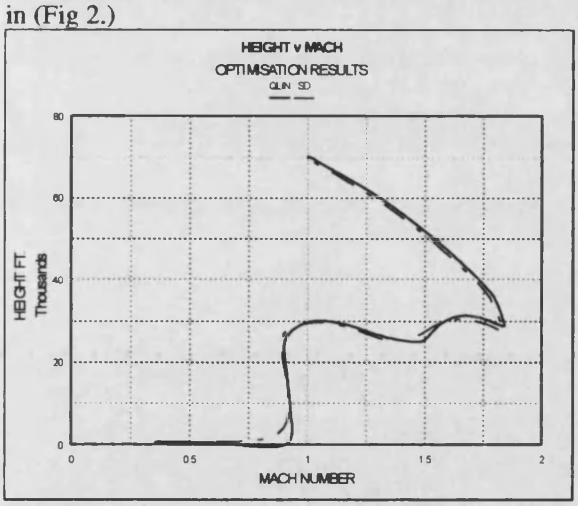


FIGURE 2.

From the results obtained in the optimisation process the nominal pitch attitude time history was computed from the optimum time profiles for flight path angle and angle of attack . Differentiation of this generated a desired pitch rate profile to be followed in order to fly the optimum manoeuvre. This signal was used as the excitation for the combined C.S.A.S. aircraft system to investigate the on-line identification of the aircraft parameters.

Parameter Variations on the Optimal Trajectory

Throughout the optimum trajectory at every time step in the integration procedure the parameters of the small perturbation pitch rate per elevator transfer function were computed.

$$\frac{q}{\eta} = \frac{K_0\omega_n^2(1+sT)}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

The variation of the d.c. gain, natural frequency, damping ratio and lead time constant are shown with respect to dynamic pressure and Mach number in (Figs. 3-6)

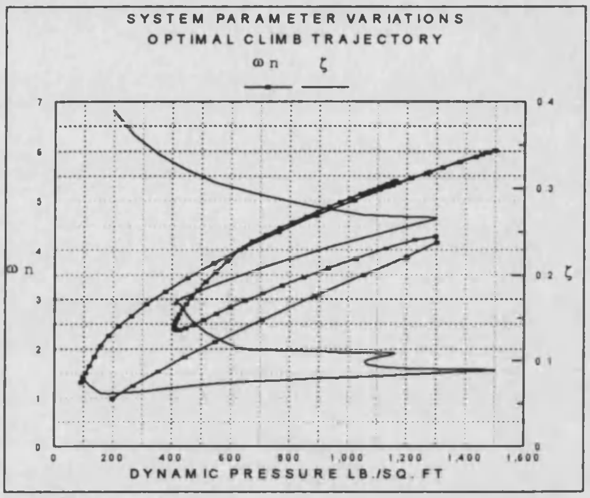


FIGURE 3

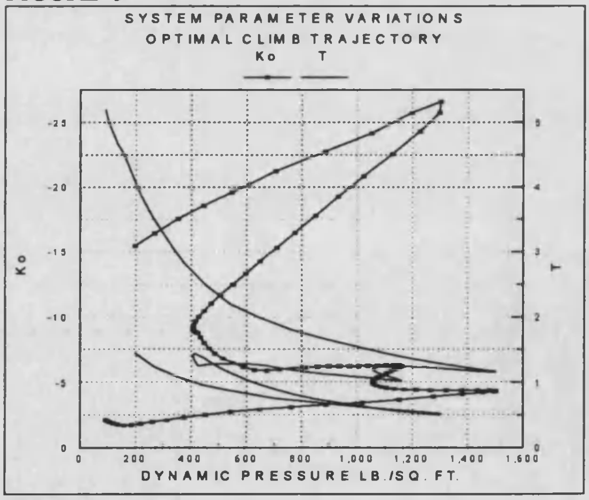


FIGURE 4

It is readily seen that during the optimal trajectory the d.c. gain K_0 varies by a ratio of approximately ten to one, the natural frequency by six to one and the lead time constant by ten to one, while the damping ratio varies from a value of 0.4 at the start of the trajectory down to about 0.06 towards the end. It is also seen that the parameter variations are very non linear and not single-valued . This increases the complexity of devising a simple gain scheduling algorithm for the C.S.A.S. .

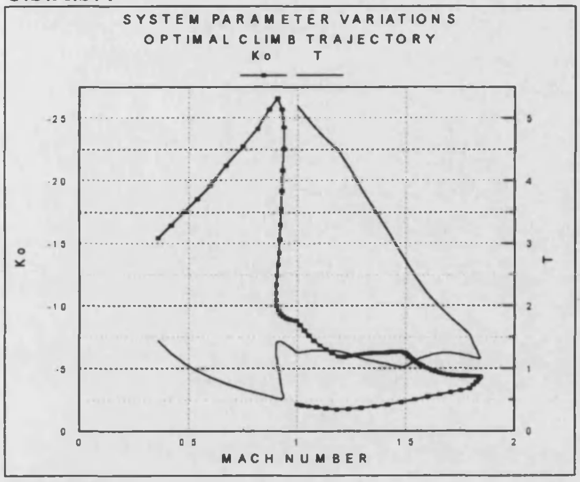


FIGURE 5.

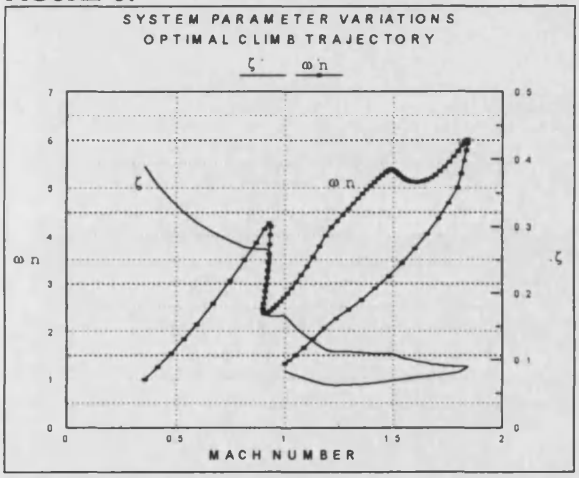


FIGURE 6.

Because of these difficulties an on-line adaptation of the C.S.A.S. with respect to continuously identified parameters was considered.

C.S.A.S. Structure.

A simple proportional plus lagged integral controller of the form of (1) was chosen as the C.S.A.S. for the purpose of the investigation.

$$\frac{\eta_D(s)}{q_{ERROR}(s)} = K_c\{1 + \frac{K_i}{s(1 + sT_c)}\} \qquad (1)$$

The relationship between controller and aircraft short period parameters used in this study is defined as

$$K_c \propto \frac{2\xi}{K_0\omega_n}, \quad T_c = \frac{1}{2\xi\omega_n}, \quad K_i = T_c\omega_n^2$$

where the constant of proportionality on K_c is selected to give the desired break frequency of the combined aircraft and C.S.A.S. closed loop response characteristic. For the purpose of the investigation this was set at 4 rad./sec. To on-line adapt the C.S.A.S. parameters, the aircraft parameters including the lead time constant were identified and tracked during the optimal trajectory. The envelope of transient closed loop pitch rate step responses of the adapted C.S.A.S. aircraft system is shown in (Fig 7) and the uniformity of response obtained is clearly demonstrated. The small variation in this envelope is caused by the ratio of the aircraft lead-time constant to the controller lag- time constant . For the purpose of comparison the envelope of transient responses obtained with a set of fixed parameter settings for the C.S.A.S. is shown in (Fig. 8). The controller parameters were set to the mid point of their adaptive range in this exercise. The spread of response is evident in both natural frequency and damping ratio and at some flight cases on the climb trajectory the system is unstable. Investigations have indicated that this response characteristic could be improved by scheduling the integrator gain as a function of Mach, however this is not exact and so the on-line identification and adaptation procedure has been implemented.

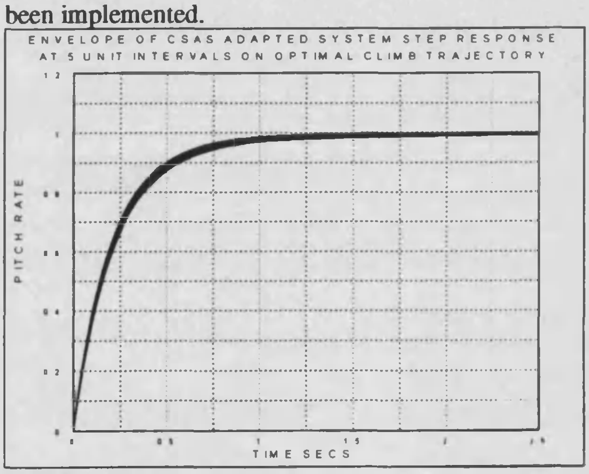


FIGURE 7

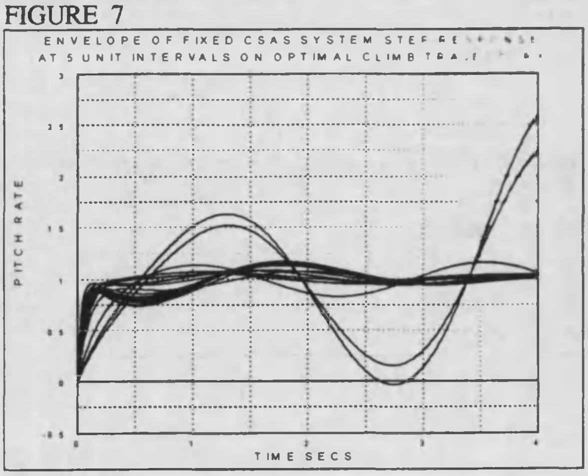


FIGURE 8

On-line Parameter Identification And Tracking.

The method of on-line identification selected is that of quasilinearisation[4, 5,6]. In general a non-linear system of the form $\dot{\underline{x}} = f(\underline{x}, \underline{u}, \underline{k}, t)$ where $\underline{k}$ represents the unknown time varying parameter set, is linearised using the Newton-Raphson algorithm

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{k}} \end{bmatrix}_{n+1} = [J(\underline{x}_n ; \underline{k}_n)] \begin{bmatrix} \underline{x}_{n+1} - \underline{x}_n \\ \underline{k}_{n+1} - \underline{k}_n \end{bmatrix} + f(\underline{x}_n, \underline{u}_n, \underline{k}_n, t)$$

Although the actual system parameters are time varying, it is assumed for the purpose of identification that during the short time periods required for identification they are constant. At the end of each identification the best-fit constant values for the parameters is determined. The identification process is continuous and in this manner the unknown time varying parameters are tracked as piece-wise constant values. The results are similar to a discrete sampling of the time varying parameters. In the case under consideration of the identification of the short period dynamics of the aircraft four unknown parameters are required to be identified. Re-defining the aircraft system as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -k_1 & 1 \\ -k_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} k_3 \\ k_4 \end{bmatrix} u$$

with x_1 representing the pitch rate of the aircraft and u as the elevator input from the C.S.A.S. controller, the unknown parameters are now k_1 , k_2 , k_3 , k_4 where

$k_1 = 2\xi\omega_n$, $k_2 = \omega_n^2$, $k_3 = K_0 \omega_n^2 T$, $k_4 = K_0 \omega_n^2$

The linearised equations become

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{k}_1 \\ \dot{k}_2 \\ \dot{k}_3 \\ \dot{k}_4 \end{bmatrix}_{N+1} = \begin{bmatrix} -k_{1N} & 1 & -x_{1N} & 0 & u_N & 0 \\ -k_{2N} & 0 & 0 & -x_{1N} & 0 & u_N \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{1N+1} - x_{1N} \\ x_{2N+1} - x_{2N} \\ k_{1N+1} - k_{1N} \\ k_{2N+1} - k_{2N} \\ k_{3N+1} - k_{3N} \\ k_{4N+1} - k_{4N} \end{bmatrix} + \begin{bmatrix} -k_{1N} x_{1N} + \dot{x}_{2N} + k_{3N} u_N \\ -k_{2N} x_{1N} + k_{2N} u_N \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The identification task is now a boundary value problem where the initial conditions on the unknown parameters have to be selected such that the states of an identification model take on those of the aircraft states during the identification interval. As there are twice as many unknown parameters as there are states, two points in the identification interval are chosen , namely the mid point and end point of the interval from which the unknown initial conditions of the parameters are computed. The identification procedure commences with selecting a set of starting vectors for the coefficients of the Jacobian matrix. The obvious choice for initialising the iteration procedure is to use actual measurements of the aircraft system states and actual control input to the system. Starting vectors for the four unknown parameters are chosen as $k_i = g_i$ where the g_i are constants equivalent to the mid point of the range of the individual parameters. Expanding the linearised equations and making the above substitutions, the linearised equations become

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{k}_1 \\ \dot{k}_2 \\ \dot{k}_3 \\ \dot{k}_4 \end{bmatrix}_{N+1} = \begin{bmatrix} -g_1 & 1 & -x_{1s} & 0 & u_s & 0 \\ -g_2 & 0 & 0 & -x_{1s} & 0 & u_s \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{1N+1} \\ x_{2N+1} \\ k_{1N+1} \\ k_{2N+1} \\ k_{3N+1} \\ k_{4N+1} \end{bmatrix} + \begin{bmatrix} g_1 x_{1s} \\ g_2 x_{1s} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The solution of these equations consists of a particular integration obtained with initial conditions on the unknown parameters of $k_i(0) = g_i$, together with a linear combination of ,in this instance four, sets of homogeneous solutions. The overall time solution during the identification interval is given by

$$\begin{bmatrix} x_{1m}(t) \\ x_{2m}(t) \\ k_1(t) \\ k_2(t) \\ k_3(t) \\ k_4(t) \end{bmatrix} = C_1 \begin{bmatrix} x_{1H1}(t) \\ x_{2H1}(t) \\ k_{1H1}(t) \\ k_{2H1}(t) \\ k_{3H1}(t) \\ k_{4H1}(t) \end{bmatrix} + C_2 \begin{bmatrix} x_{1H2}(t) \\ x_{2H2}(t) \\ k_{1H2}(t) \\ k_{2H2}(t) \\ k_{3H2}(t) \\ k_{4H2}(t) \end{bmatrix} + C_3 \begin{bmatrix} x_{1H3}(t) \\ x_{2H3}(t) \\ k_{1H3}(t) \\ k_{2H3}(t) \\ k_{3H3}(t) \\ k_{4H3}(t) \end{bmatrix} + C_4 \begin{bmatrix} x_{1H4}(t) \\ x_{2H4}(t) \\ k_{1H4}(t) \\ k_{2H4}(t) \\ k_{3H4}(t) \\ k_{4H4}(t) \end{bmatrix} + \begin{bmatrix} x_{1PI}(t) \\ x_{2PI}(t) \\ k_{1PI}(t) \\ k_{2PI}(t) \\ k_{3PI}(t) \\ k_{4PI}(t) \end{bmatrix}$$

EQUATION 2

It should be noted that the particular integration system as defined is a mathematical model of the aircraft system having an identical structure but with estimates for the unknown parameters. As the choice of the parameters are only estimates they will initially be incorrect and the model responses will not match the actual system state responses. These therefore have to be corrected by the linear combinations of homogeneous integrations. The initial conditions of the states of the model for the particular integration are set to the values of the actual system states pertaining at the start of the identification interval. The initial conditions for each set of homogeneous solution are defined as follows and are specifically chosen to simplify subsequent computation.

$$\begin{bmatrix} x_{1Hi} \\ x_{2Hi} \\ k_{1Hi} \\ k_{2Hi} \\ k_{3Hi} \\ k_{4Hi} \end{bmatrix}_{i=0} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}_{i=1} ; \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}_{i=2} ; \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}_{i=3} ; \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}_{i=4}$$

The complete sets of homogeneous integrations and the particular integration are computed simultaneously from the overall system of equations defined in (3).

The four weighting constants C_i of the sets of homogeneous solutions in (2) are computed from (4) where t_1 is the mid point of the identification interval and t_f is the end point.

$$\begin{bmatrix} \dot{x}_{1u} \\ \dot{x}_{2u} \\ \dot{x}_{1PJ} \\ \dot{x}_{2PJ} \\ \dot{x}_{1H1} \\ \dot{x}_{2H1} \\ \dot{x}_{1H2} \\ \dot{x}_{2H2} \\ \dot{x}_{1H3} \\ \dot{x}_{2H3} \\ \dot{x}_{1H4} \\ \dot{x}_{2H4} \end{bmatrix} = \begin{bmatrix} -k_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -k_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -g_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -g_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -g_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -g_2 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & -g_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -g_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g_1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g_1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g_2 & 0 \end{bmatrix} \begin{bmatrix} x_{1u} \\ x_{2u} \\ x_{1PJ} \\ x_{2PJ} \\ x_{1H1} \\ x_{2H1} \\ x_{1H2} \\ x_{2H2} \\ x_{1H3} \\ x_{2H3} \\ x_{1H4} \\ x_{2H4} \end{bmatrix} + \begin{bmatrix} k_3 u_e \\ k_4 u_e \\ g_3 u_e \\ g_4 u_e \\ -x_u \\ 0 \\ 0 \\ -x_u \\ u_e \\ 0 \\ 0 \\ u_e \end{bmatrix}$$

EQUATION 3.

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix} = \begin{bmatrix} x_{1H1}(t_1) & x_{1H2}(t_1) & x_{1H3}(t_1) & x_{1H4}(t_1) \\ x_{2H1}(t_1) & x_{2H2}(t_1) & x_{2H3}(t_1) & x_{2H4}(t_1) \\ x_{1H1}(t_f) & x_{1H2}(t_f) & x_{1H3}(t_f) & x_{1H4}(t_f) \\ x_{2H1}(t_f) & x_{2H2}(t_f) & x_{2H3}(t_f) & x_{2H4}(t_f) \end{bmatrix}^{-1} \begin{bmatrix} x_{1u}(t_1) - x_{1PJ}(t_1) \\ x_{2u}(t_1) - x_{2PJ}(t_1) \\ x_{1u}(t_f) - x_{1PJ}(t_f) \\ x_{2u}(t_f) - x_{2PJ}(t_f) \end{bmatrix}$$

EQUATION 4.

The corrected or identified values for the unknown parameters during the identification interval then become $k_i = C_i + g_i$.

As the small perturbation representation of the aircraft system dynamics is linear, convergence to the correct values of the identified parameters is single step and it is unnecessary to iterate the procedure to obtain convergence. If however a non-linear representation of aircraft dynamics had been chosen it would have been necessary to perform several iterations before a satisfactorily-converged identification is achieved. This would involve an extension of the lapse time to achieve identification; however this situation can be alleviated by time scaling equations (3) for subsequent iterations after the first, which of necessity is computed in real time in synchronism with the actual aircraft response. The identified values of the system parameters so determined by the above procedure become the starting estimates of the unknown parameters for the next identification interval. In this manner the identification process is continuous and the time varying parameters are tracked as constants during each identification interval. Any identification process requires persistency of excitation and this can be checked at each identification step by determining that the matrix of homogeneous solutions in (4) is non-singular. Should this prove not to be the case the values of the identified parameters are held at the last-identified values until persistency of excitation resumes. The results of the identification and tracking of the four aircraft parameters while being controlled by the adaptive system on the optimum climb trajectory are shown in (Figs 9-10). The actual aircraft parameters and the identified parameters are superimposed on each other and the accuracy of the identification and tracking of all four system parameters is evident.

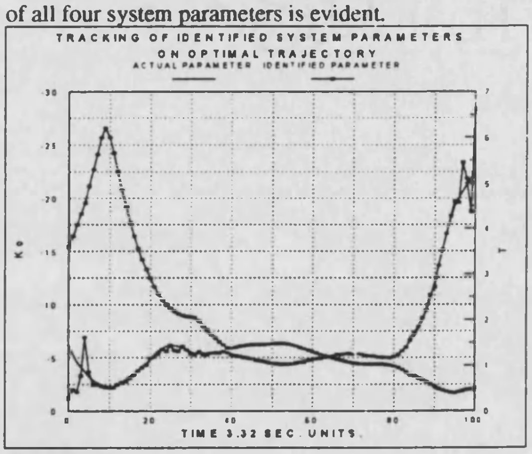


FIGURE 9

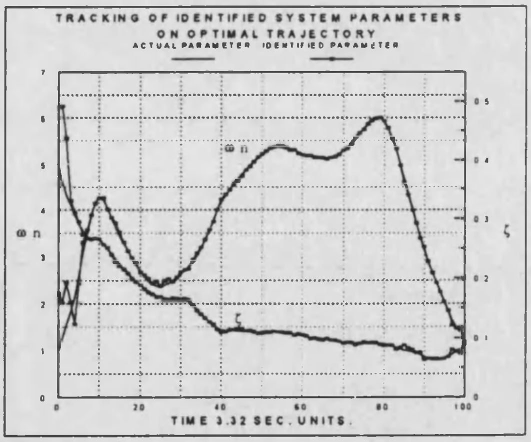


FIGURE 10

On-Line Adaptation Of C.S.A.S

At the end of each identification interval the controller parameters are updated as a function of the identified aircraft parameters. The controller parameter variations on the optimal climb trajectory are as shown in (Figs 11- 12). Also shown is the ratio of the aircraft lead time constant to the controller lag time constant throughout the manoeuvre. Although these are not matched exactly the deviation in this ratio from unity is small and the effect on the envelope of closed-loop aircraft /C.S.A.S. combination is negligible. By this technique the handling qualities of the aircraft in the pitch axis are maintained virtually uniform on the optimal climb trajectory. For the purpose of this investigation the identification interval was fixed at 200ms. which proved satisfactory. It would be perfectly feasible to select the identification interval as a function of the identified natural frequency of the system.

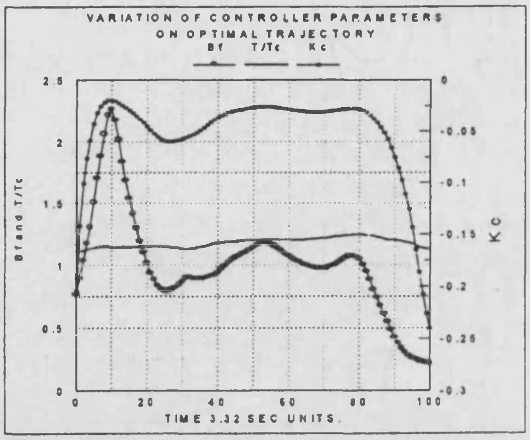


FIGURE 11

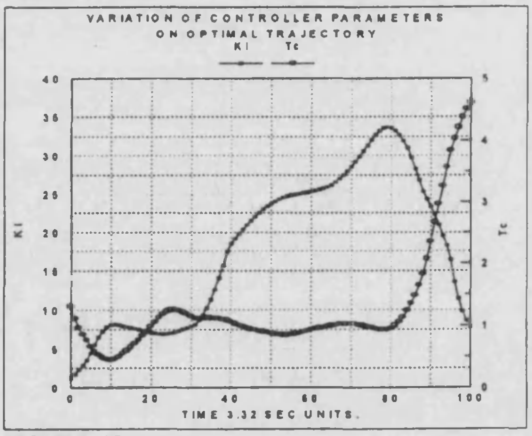


FIGURE 12

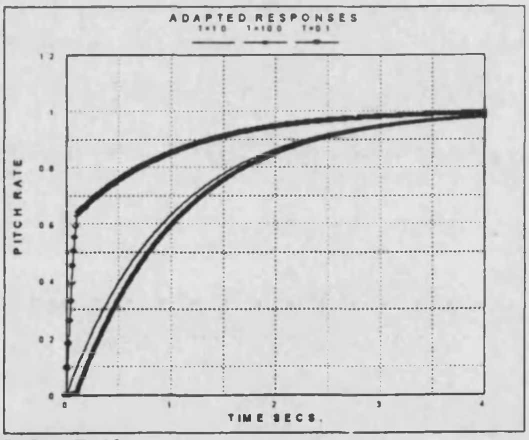


FIGURE 13

Augmented Control

As the identification and adaptation is occurring within the transient response time of the system it is interesting to investigate this further as a separate exercise. Starting with a nominal transient response characteristic for the closed-loop pitch rate response, then during the initial identification period and before identification and adaptation have occurred the actual response deviates from the nominal. This is due to the mismatch of the C.S.A.S. with the as yet unidentified aircraft. On adaptation the transient response continues from this point onwards with the now correct dynamics. This satisfies the handling criterion; however the actual transient response characteristic deviates from the nominal. (Fig 13). If the requirement is to minimise the error between the nominal and actual transient trajectory this may be done from the point of identification onward by minimising a quadratic function of this error subject to the now known dynamic constraints of the system. This optimisation would normally be performed off-line; however since an on-board tracking model of the aircraft exists this could be used to perform the optimisation and on-line generation of the additional augmented feed back control. The effect of applying this augmented control at the point of adaptation is shown in (Fig 14). Here two mismatched controllers are considered which represent a deviation by a factor of 100 from the nominal dynamics. The system response is forced back onto the nominal transient trajectory, by the augmented control, from the point of adaptation onwards. The resultant accelerations produced can be controlled by the introduction of state constraints. (Fig 15) shows that there is some benefit in using this optimal augmented control even during the initial identification period as the excursions away from the nominal transient response are significantly reduced.

Conclusion

The variation of aircraft parameters on an optimal manoeuvre has been investigated and it has been shown that the design of a parameter scheduled controller using auxiliary variables is complex. An on- line identification and tracking model has been implemented using the normal commands to perform the manoeuvre, and without resorting to additional test signals for the purpose of parameter identification. Functions of the identified values of the aircraft parameters have been used to adapt the controller, in order to provide a virtually uniform response characteristic throughout the manoeuvre. The adaptation is continuous and operates within the transient response time of the system. An optimum controller augmentation has been studied which uses the identified parameter information to compensate for deviations from a nominal transient response arising during the identification interval. The augmented control returns the transient response trajectory to the nominal in an optimum manner. This augmented control also reduces deviations from the nominal transient response when there is mismatch between the controller and aircraft dynamics during the initial identification interval.

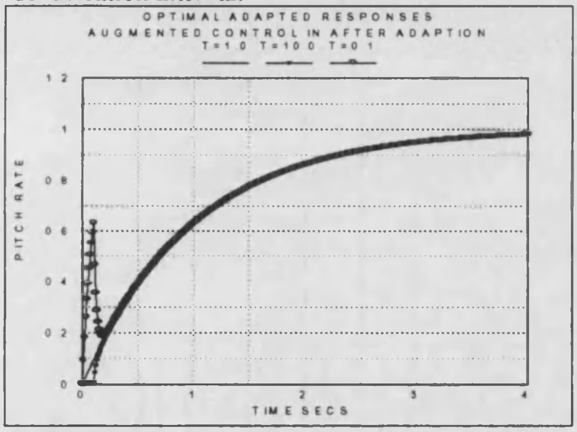


FIGURE 14

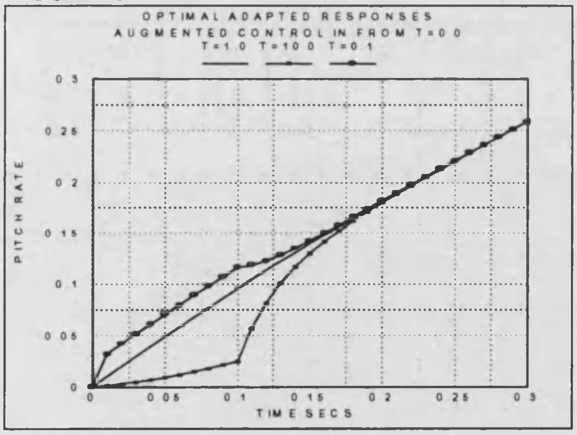


FIGURE 15

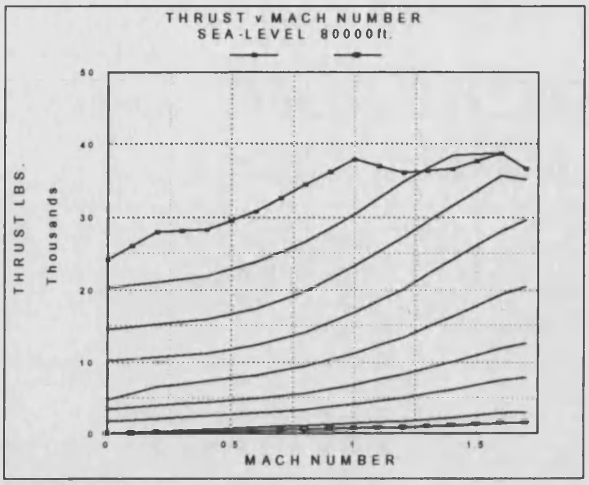


FIGURE 16

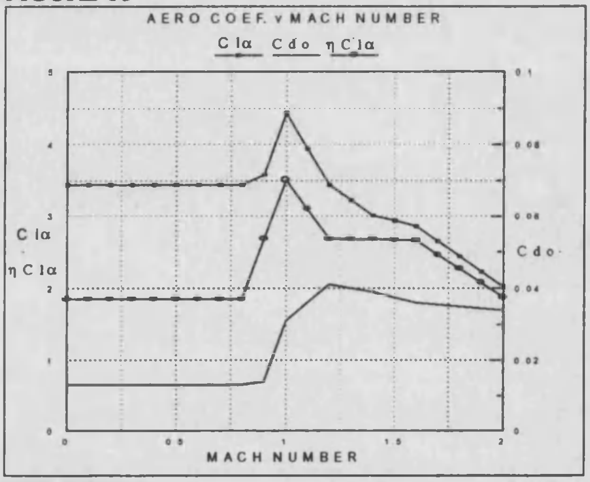


FIGURE 17

References

1. Bryson. A.E., Desai M.N. and Hoffman. 1969. Energy state approximation in performance optimisation of supersonic aircraft. J. Aircraft 6:481-8
2. Bryson. A.E. and Yu-Chi Ho 1975 Applied Optimal Control., Hemisphere Publishing Corporation.
3. Kirk D.E. 1970. Optimal Control Theory. Prentice-Hall.
4. Bellman R.E. and Kalaba R.E., 1965 Quasilinearisation and Non-linear Boundary-Value Problems., Elsevier Publishing Company, Ltd.,
5. Mc Gill R. and Kenneth P., 1964. Solution of Variational Problems by means of the Generalised Newton-Raphson Operator. A.I.A.A. Journal.,Vol 2.
6. Kumar K.S. and Sridhar R. 1964., On the Identification of Control Systems by the Quasi-Linearization Method., I.E.E.E. Trans. Automatic Control 152-4.
7. MacCormac J.K.M. 1968., The use of hybrid computation in an on-line identification scheme. IFAC, Budapest.
8. Mc.Lean D. Automatic Flight Control Systems.,1990. Prentice-Hall.
9. Soderstrom T. and Stoica P. 1989. System Identification., Prentice-Hall.

THE USE OF HYBRID COMPUTATION IN AN ON-LINE IDENTIFICATION SCHEME

J.K.M. MacCormac

Electrical Engineering Department
University College of Swansea
Glamorgan, S.Wales, U.K.

Introduction

An optimal control law for a non-linear system may be derived using Calculus of Variations. Subject to dynamical constraints a performance index is minimised while specified initial and terminal conditions are satisfied. The resultant control law will provide nominal state variable trajectories. It is however an open-loop form of control. Changes in the initial and terminal conditions require a recomputation to determine the new optimal control.

A method to reoptimise the control about a reference, without recomputation, has been described in [1]. Based on the Second Variation this scheme provides a linear feed-back control with time varying coefficients. The reference control is continuously modified, to give neighbouring optimal control. Signals proportional to the small deviations of the actual response from the nominal are used to augment the reference control.

When the dynamics of a system are changing with environment the reference control will not be optimal at different operating conditions. In this case it is necessary to repeatedly update the control to match the current dynamical state. A combination of this updating procedure and the neighbouring optimal scheme has the following advantage. A new reference control may be computed while small disturbances about the nominal responses are being accounted for by the previous computed feedback control. Figure 1 is a block diagram representation of this scheme.

To perform this recomputation of the reference control it is necessary to have a knowledge of the current state of the dynamics. This is achieved by on-line identification.

Optimal Identification

To preserve overall system optimality it is desirable to introduce some degree of optimisation into the identification process. This may be done again using a variational approach.

A mathematical model of the non-linear system may be represented by the vector differential equation.

$$\dot{x} = f(x,p,t) \qquad \ldots (1)$$

where $x^T = (x_1, x_2 \ldots x_n)$

and $p^T = (p_1, p_2 \ldots p_m)$

are respectively the state variables and unknown parameters. It is assumed the structure of the system is known. 'f' is the n-dimensional vector function representing this structure. Also available are a set of system state variable and control histories for the interval $t_o < t < t_1$.

The model is excited by the recorded system control signals. The identification problem is now one of minimising some error function between model and system responses with respect to the model parameters. A quadratic performance index of the form shown in (2) is chosen.

$$P = \phi(p)_{t=t_o} + \int_{t_o}^{t_1} g(x-x_s)dt \qquad \ldots (2)$$

where $\phi(p) \equiv p^T \frac{A}{2} p$

$g(x-x_s) \equiv (x-x_s)^T \frac{K}{2}(x-x_s)$

A and K are both weighting matrices. K gives the cost of computed model res-

responses x differing from the system responses x_s. If the system recordings are contaminated by noise K may be the inverse of the variance matrix of independent noise on the different state variables.

The performance index must be minimised subject to the model dynamic constraints (1) and the boundary conditions on the measured state variables. For the purpose of this paper the system parameters are assumed constant over the short recording interval. An additional constraint then is $\dot{p} = o$ ... (3)

Adjoin the constraints (1) and (3) to the performance index with Lagrange multipliers λ and μ.

$$J = \phi(p)_{t=t_o} + \int_{t_o}^{t_1} \Big[g(x-x_s) + \lambda^T(\dot{x}-f(x,p,t)) + \mu^T \dot{p} \Big] dt \qquad \ldots (4)$$

The first variation [2] of J is

$$\delta J = \nabla p\,(\phi)^T\,\delta p \Big|_{t=t_o} + \Big|\lambda^T \delta x + \mu^T \delta p\Big|_{t_o}^{t_1}$$

$$+ \int_{t_o}^{t} \Big[(\nabla x(g)^T - \dot{\lambda}^T - \lambda^T Jx|f|)\delta x - (\lambda^T Jp|f| + \dot{\mu}^T)\delta p \Big] dt \qquad \ldots (5)$$

where in general

$$\nabla y\,(a)^T = \left[\frac{\partial a}{\partial y_1}, \frac{\partial a}{\partial y_2}, \ldots \frac{\partial a}{\partial y_n}\right]$$

Specifically $\nabla p(\phi) = p^T A$

$\nabla x(g) = (x-x_s)^T K$

$J_y|f|$ is a general Jacobian matrix with elements jk $\dfrac{\partial f_j}{\partial y_k}$

The necessary conditions for an extremum of J over the interval $t_o < t < t_1$ are

$$\left.\begin{aligned} \dot{x} &= f(x,p,t) \\ \dot{p} &= o \\ \dot{\lambda} &= -(J_x|f|)^T \lambda + (x-x_s)^T K \\ \dot{\mu} &= -(J_p|f|)^T \lambda \end{aligned}\right| \qquad \ldots (6)$$

with boundary conditions

$$\left.\begin{aligned} x(t_o) &= x_o \\ \lambda(t_1) &= o \\ \mu(t_o) &= p^T A \\ \mu(t_1) &= o \end{aligned}\right| \qquad \ldots (7)$$

The solution of equation (6) such that (7) are satisfied constitutes a non-linear boundary value problem. There is no generalised method for solution of this problem. A scheme based on the Newton-Raphson Operator [3,4] provides an iterative solution.

Newton-Raphson Algorithm

The M-dimensional non-linear set of equations (6) represented by $\dot{Y} = G(Y)$ is linearised to give

$$\dot{Y}_{N+1} = J_y|G(Y_N)|\,|Y_{N+1} - Y_N| + G(Y_N) \quad \ldots (8)$$

The coefficients of the Jacobian $J_y|G(Y_N)|$ will be time varying, but they are known from the previous iterate. Given a sufficiently good approximation to the starting vector $Y_N(t)$, convergence of the sequence is monotonic and quadratic. The non-linear boundary value problem is now reduced to the modification of initial conditions for (8).

A particular solution $Y_P(t)$ is generated based on a set of starting vectors $Y_N(t)$ and initial conditions (7). Estimates $Y_E(t_o)$ are made for the m-unknown initial conditions $Y_{M-m+i}(t_o)$

2

$(i=1,2,\ldots m)$.

In general the terminal conditions $Y_p(t_f)$ will not satisfy the desired terminal conditions $Y_D(t_f)$ of (7).

m-sets of solutions $Y^j_{iH}(t)$ $((i=1,2,\ldots m), j=1,2,\ldots M)$ are obtained for the homogeneous system of equation (9).

$$\dot{Y}_{N+1} = J_y|G(Y_N)| \quad |Y_{N+1}| \qquad \ldots(9)$$

Initial conditions for the ith set are $Y_{M-m+i}(t_o) = 1$ with the other M-1 initial conditions equal to zero.

Because (8) is linear

$$Y^j_D(t_f) = \sum_{i=1}^{m} Y^j_{iH}(t_f)\, C_i + Y^j_p(t_f) \qquad (j=1,2,..M) \qquad \ldots(10)$$

Solution of (10) for $(C_1, C_2 \ldots C_m)$ gives the modification to made for the m unknown initial conditions

$$Y^j_{N+1}(t_o) = Y^j_E(t_o) + C_i \qquad \begin{matrix} j=M-m+i \\ i=1,2,..m \end{matrix} \qquad \ldots(11)$$

An integration performed now with starting vectors $Y_N(t)$, initial conditions (7) and unknown initial conditions (11) will satisfy the terminal conditions (7). This solution $Y_{N+1}(t)$ is used as the new starting vectors $Y_N(t)$ and the process repeated until convergence occurs.

Hybrid Implementation

The solution of (8) requires previous iterations $Y_N(t)$. These must be stored. Computation is needed to determine forcing functions for the particular and homogeneous integrations of the last section. It is also necessary to evaluate new estimates for the unknown initial conditions. A digital computer is well suited to perform these operations. Numerical solution however of high-order differential equations is slow compared with an analogue solution. Using an analogue and digital computer in a hybrid configuration the advantages of both may be exploited. Photograph 1 shows the SOLARTRON 247 analogue computer linked to the Digital Equipment P-D-P-8 computer.

Analogue to Digital |A/D| and Digital to Analogue |D/A| conversions are performed to 12 bit accuracy. This is equivalent to one location of core store. Floating-Point storage requires three locations per number. Interface conversions are therefore stored as integer arrays. To reduce round off error calculations are performed in floating point mode by first converting integers to this form and then reconverting to integer mode at the end of a calculation. These operations are governed by a compromise between accuracy, speed of computation and storage space.

Variables for D/A conversion must be in the range $\pm 2^n$ corresponding to $\pm$ 10 volts. It is necessary to scale these variables in floating point form before changing to integers for D/A conversion.

The digital computer also controlled the compute-hold-reset states of the analogue machine. This was done by means of a -24 volt output logic buffer.

The sequence of operations for the identification scheme is as follows

(a) Record system state variables and control signals over a short time interval.
(b) Use these state variables and estimates as starting vectors $Y_N(t)$ for step (c).
(c) PARTICULAR INTEGRATION and record $Y_p(t_f)$
(d) Homogeneous Integration and record $Y_H(t_f)$
(e) Modify estimates of unknown initial conditions $Y_E(t_o)$.
(f) Particular integration with modified initial conditions to obtain $Y_{N+1}(t)$.
(g) Replace $Y_N(t)$ by $Y_{N+1}(t)$.
(h) Iterate steps (c) through (g) until convergence test is satisfied.
(i) Repeat from step (a) for next time interval.

For on line applications it is essential that the identification time between record phases be as short as possible. To achieve this the analogue integration steps (c), (d) and (f) are performed on a faster than real time scale.

The high speed analogue model in Figure 2 represents these time scaled integrations. An additional problem then is the synchronisation between model and system such that short model integration times correspond to system real time record intervals. To overcome this the digital programme first executed a dummy model integration. At the end of this period a timing signal T was sampled. Let N be the number of samples of system response required during the record period. Then the ith sample must be taken at time $\frac{i \times T}{N}$ x (TIME SCALE FACTOR). Again this is evaluated in floating point form before conversion to integer mode for comparison with the sampled time signal. A flow chart of the digital programme is shown in figure (3).

A simple example demonstrates the use of the programme. The linearised equations for this example are

$$\dot{X}_{N+1} = -A_N X_{N+1} + (U-X_N)A_{N+1} + A_N X_N \quad ...(12)$$

$$\dot{A}_{N+1} = 0$$

Starting vectors for the state variable $X_N(t)$ are the recorded system response. U is the recorded forcing function. Boundary conditions on X are obtained at the start and finish of the record interval. An initial estimate $A_N = .5$ is chosen for the first identification. For continuous identification A_N is set to the last identified value. Traces of results obtained are shown in figure (4) through figure (6). In Figure (4) the process is running slowly to indicate the steps involved. Model traces show particular integrations followed by homogeneous integrations. The model is running on a 2:1 time scale with respect to the system and thus the response of the system is due to a unit step U.

When A has converged to the true value 1.5 the particular integration of the model corresponds to the system response during the record interval. The test for conververgence is a difference between A_N and A_{N+1} of less than one percent. When convergence occurs the procedure terminates. Figure (5) is a portion of a continuous identification of a fixed parameter. After an identification is complete a new record phase is initiated and a further identification performed. The system is being perturbed by a triangular waveform. Convergence test and model time scaling are as for figure (4). Note that the model integrations have zero initial conditions. Initial conditions are stored in the digital computer. This gives a saving in time which would be required to reset them on the analogue integrators.

Figure (6) shows a continuous identification of a time varying parameter. The system response is due to a triangular waveform and it is modulated by the variable parameter. The parameter is varied sinusoidly over the range .5 to 3. System record time was .1 sec. and the model operated on a 12:1 time scale. The identified values of the parameter are shown superimposed on the actual values.

Conclusion

The accuracy and speed of solution have shown this hybrid identification procedure to be of use in an on-line application. The approach to the problem has been made in a manner directly applicable to non-linear systems of higher order than the example. Multiparameter identification would require more homogeneous integrations to be performed per iteration. These could be done simultaneously as could the model particular and homogeneous integrations of the example. No additional time penalty would be incurred however more interface facilities would be required. The complete time between successive identifications would be greatly reduced if recording of the system was taking place while the model iterations are being performed. At the end of one identification it would then not be necessary to wait for the next recording interval to elapse before starting the next identification. Continuous recording may be done if a programme interrupt facility is available for the digital computer.

References

1. OPTIMISATION AND CONTROL OF NON-LINEAR SYSTEMS USING THE SECOND VARIATION. John V. Breakwell; Jason E.L. Speyer and Arthur E. Bryson. J.S.I.A.M. Control. 1963.

2. LECTURES ON THE CALCULUS OF VARIATIONS. G.A. Bliss. The University of Chicago Press, Chicago 1946.

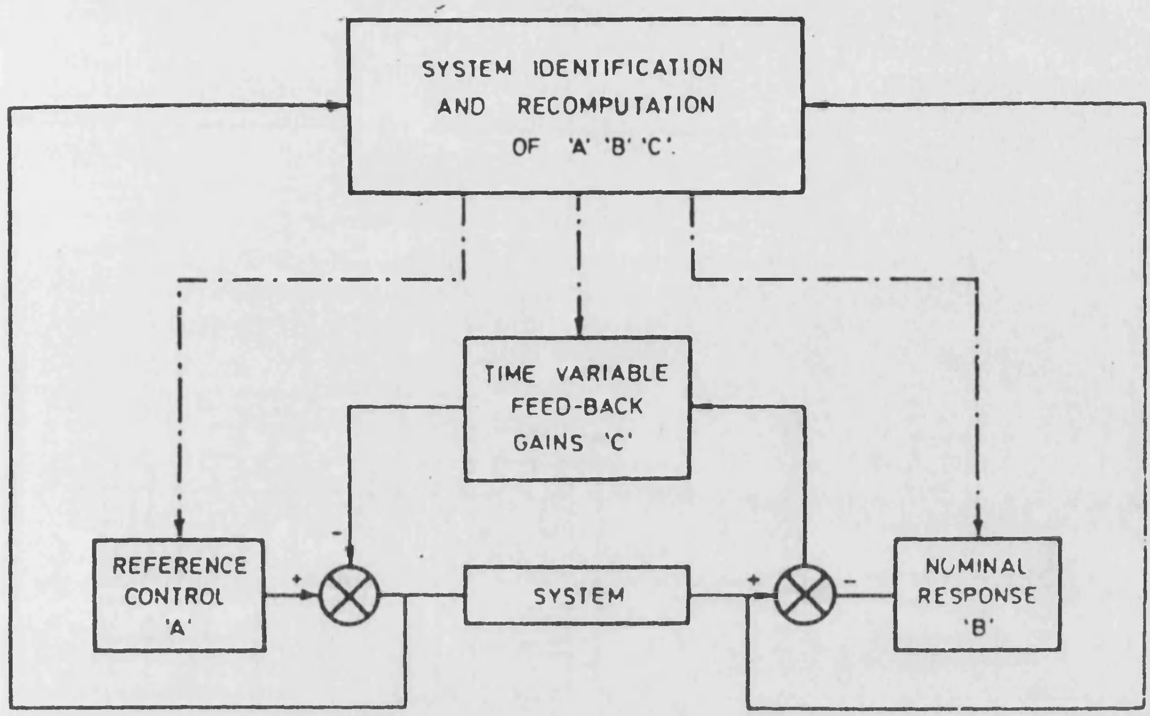

SYSTEM IDENTIFICATION
AND RECOMPUTATION
OF 'A' 'B' 'C'.
TIME VARIABLE
FEED-BACK
GAINS 'C'
REFERENCE
CONTROL
'A'
SYSTEM
NOMINAL
RESPONSE
'B'
+
-
+
-

FIG. 1.

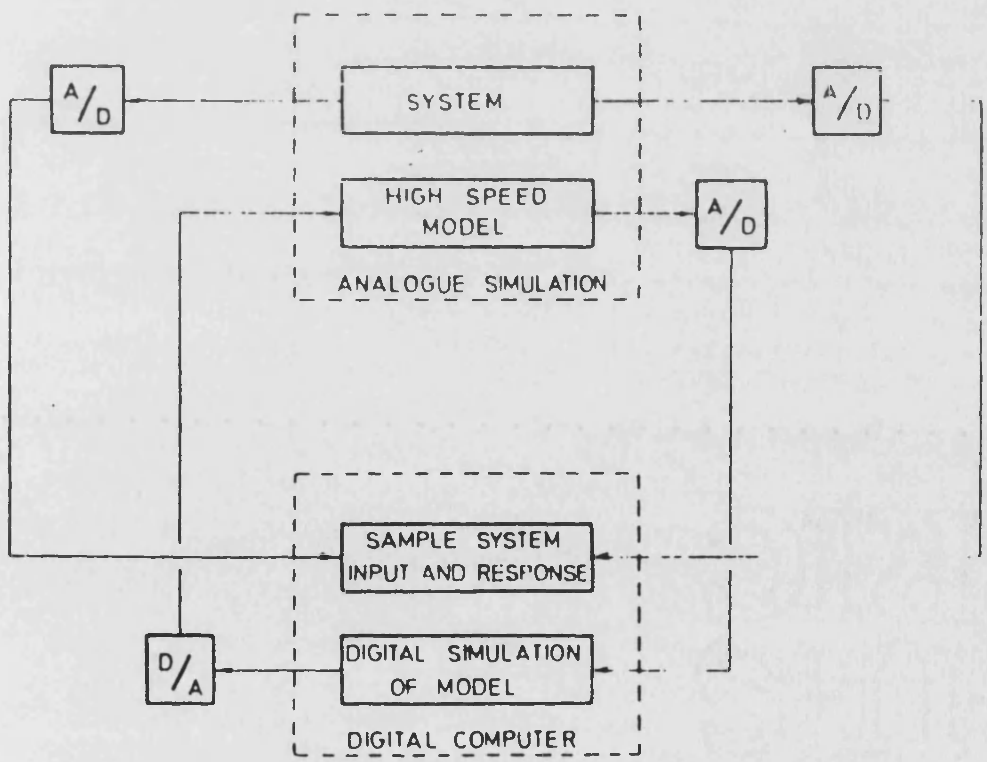

A/D
SYSTEM
A/D
HIGH SPEED
MODEL
A/D
ANALOGUE SIMULATION
SAMPLE SYSTEM
INPUT AND RESPONSE
D/A
DIGITAL SIMULATION
OF MODEL
DIGITAL COMPUTER

FIG 2

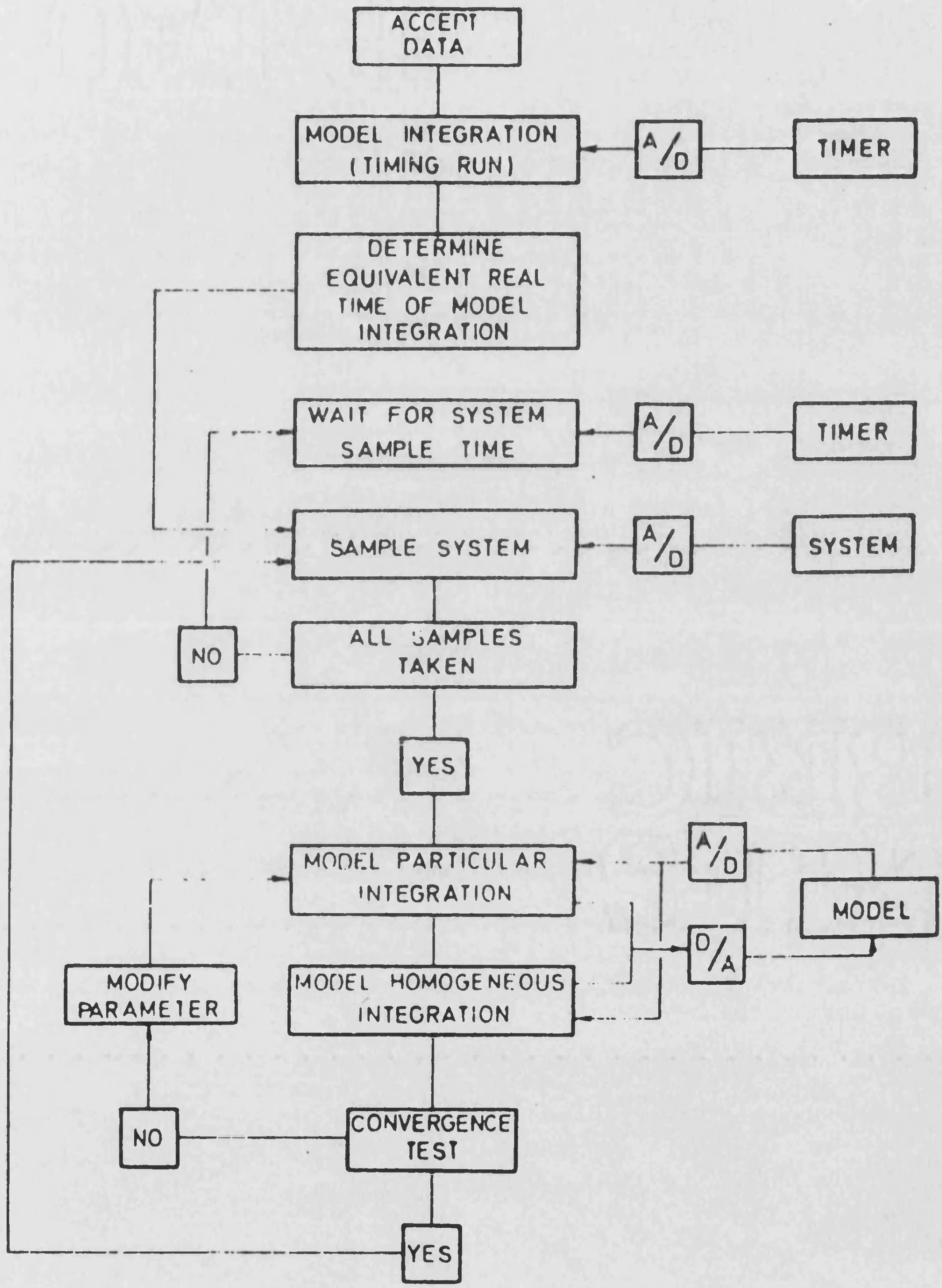

ACCEPT DATA
MODEL INTEGRATION (TIMING RUN)
A/D
TIMER
DETERMINE EQUIVALENT REAL TIME OF MODEL INTEGRATION
WAIT FOR SYSTEM SAMPLE TIME
A/D
TIMER
SAMPLE SYSTEM
A/D
SYSTEM
NO
ALL SAMPLES TAKEN
YES
MODEL PARTICULAR INTEGRATION
A/D
MODEL
D/A
MODIFY PARAMETER
MODEL HOMOGENEOUS INTEGRATION
NO
CONVERGENCE TEST
YES

FIG 3

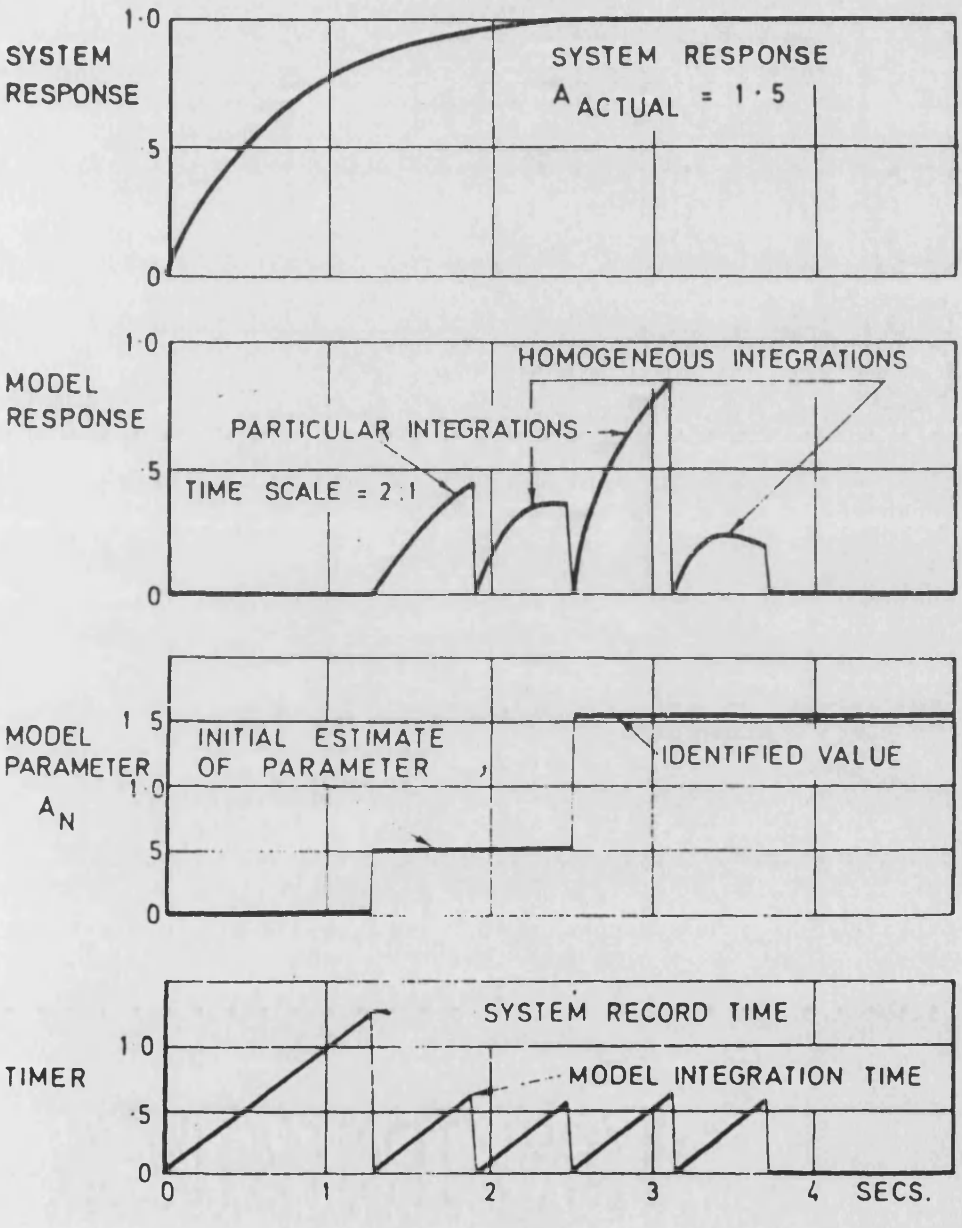

SYSTEM RESPONSE
SYSTEM RESPONSE
A_{ACTUAL} = 1·5
MODEL RESPONSE
HOMOGENEOUS INTEGRATIONS
PARTICULAR INTEGRATIONS
TIME SCALE = 2:1
MODEL PARAMETER A_N
INITIAL ESTIMATE OF PARAMETER
IDENTIFIED VALUE
TIMER
SYSTEM RECORD TIME
MODEL INTEGRATION TIME
1·0
·5
0
1 5
5
0
1
2
3
4
SECS.

FIG.

SUMMARY

Optimal self-adapting systems are employed in the control of processes, to compensate for deviations in plant response from the nominal. These methods make use of known plant parameters. For processes operating under changing environmental conditions, it is necessary to determine the current state of the plant parameters. The paper presents a Hybrid Computer solution to this identification problem.

Identification is achieved using a Variational Approach. The minimisation of a functional, providing the cost of deviations of a mathematical model from the system is performed. Account of measurements of process state in the presence of noise is included in the cost criterion. Solution of the Non-Linear Boundary Value Problem associated with Variational methods is obtained using the generalised Newton-Raphson technique. Hybrid implementation is employed to obtain the speed of identification necessary for on-line application. Problems relating to hybrid computational procedures are discussed.

ПРИМЕНЕНИЕ ГИБРИДНОГО ВЫЧИСЛЕНИЯ В СХЕМЕ ИДЕНТИФИКАЦИИ В ИСТИННОМ МАСШТАБЕ ВРЕМЕНИ

К.М. МэкКормек

(О.К.)

Р е з ю м е

Для компенсации отклонений характеристик объекта от номинальных при управлении процессами применяются оптимальные самонастраивающиеся системы. Эти методы используют известные параметры объектов. Для процессов при изменяющихся внешних условиях необходимо определить текущие значения параметров объекта. В статье представляется решение этой проблемы идентификации с помощью гибридного вычислителя.

Идентификация реализуется применением вариационного подхода. Производится минимализация функционала выражающего цену отклонения математического моделя от реалной системы. Результаты измерений состояния процесса при наличии шума включены в критерий цены. Применением обобщённой техники Ньютона-Рафсона получается решение проблемы нелинейного предельного значения объединённой вариационным методом. Для достижения необходимой скорости идентификации в истинном масштабе времени применяется гибридный вычислитель. Освещаются проблемы связанные с гибридными вычислительными процедурами.

UNIVERSITY OF WALES SWANSEA

ELECTRICAL ENGINEERING

MINISTRY OF TECHNOLOGY CONTRACT NO. PD/81/03

A Report on work on Sampled Data Adaptive Aircraft Control Systems.

by

J.K.M. MacCormac.

University College of Swansea.

ABSTRACT

This report provides the mathematical basis for optimum adaptive aircraft control systems. The general approach to the problem, incorporating relevant theory is out-lined in Section 1.

Section 2 describes by way of an example, the solution of a specific problem arising out of application of optimisation techniques which are based on the calculus of variations.

An On-line Hybrid Identification Process has been implemented to investigate the feasibility of employing techniques developed in Sections 1 and 2. This work is the subject of Section 3.

Section 4 concludes this report with an interpretation of results.

Topics of research related to this report - currently being investigated by the author - are also out-lined.

SECTION 1.

MATHEMATICAL THEORY OF AN OPTIMUM ADAPTIVE PROCESS

Introduction.

Reference (1) describes how a control law, which has been designed to produce an optimum response, may be continuously modified to account for small deviations of the actual response from the nominal. This reoptimisation procedure is based on the Calculus of Variations (2) and makes specific use of the second variation to obtain a linear time varying control law. The method is applicable to non-linear systems, however it is assumed the dynamics of the system, are known.

In adaptive control of aircraft the problem may be considered in the same manner in the sense that it is necessary to modify a control law such that the actual flight trajectory closely follows some desired nominal response. Because the dynamics of the aircraft are continuously changing with environment, they are not known a priori and hence before a technique similar to (1) may be employed it is necessary to determine the current state of the dynamics of the system. This on line determination of system dynamics is simplified by a knowledge of the structure or order of the dynamics as is available in the case of aircraft.

It is desirable in order to maintain overall system optimality to introduce some degree of optimisation into the parameter identification process. This may be achieved again using a variational approach thus enabling the re-optimisation of the problem, to be performed as an extension of the identification procedure; the mathematical theory underlying both phases, being intimately related.

Consider in general terms the principles of optimisation using variation methods. This is instructive as it highlights a specific problem common to optimisation procedures. It is the method of solution of this problem which forms the basis of this paper and also serves as a pointer to the feasibility of employing this particular method of identification in aircraft adaptive control systems.

Optimisation involves the minimisation of a functional - the pay off criterion - with respect to the variables it is desired to optimise.

For problems in optimal control the cost function has the form

$$F(U) = g(t_o, X(t_o); t_1, X(t_1)) + \int_{t_o}^{t_1} f(t, X, U)dt \qquad \text{... 1.1.}$$

where it is desired to determine the R-dimensional vector

$$U = (U_1(t), U_2(t) \ldots U_R(t)) \quad t_o \leq t \leq t_1 \qquad \text{... 1.2.}$$

such that the best pay off is obtained subject to the following constraints being satisfied.

(a) The state variables

$X = (X_1(t), X_2(t) \ldots X_N(t))$ satisfy a set of differential equations - namely the dynamics of the system.

$\dot{X}_i = G_i(t,X,U) \qquad i = 1,2, \ldots N \qquad \ldots 1.3.$

i.e. $\phi(t,X,\dot{X},U) = 0$

(b) Prescribed initial and terminal conditions are required to be satisfied

$o_i(t_o,X(t_o); t_1,X(t_1)) = 0 \qquad i = 1,2, \ldots p \qquad \ldots 1.4.$

(c) In addition it may be necessary to satisfy inequality constraints on either the control vectors or state variables - implying that these must not exceed certain magnitudes.

$P_i(t,X,U) \geq 0 \qquad i = 1,2, \ldots m \qquad \ldots 1.5.$

Inequality constraints introduce additional complexity. ~~References~~ Reference (~~3,4~~,5) indicates how they may be included in a manner which pursues uniformity of approach to the optimisation problem.

For the cost function to be a minimum the following necessary conditions must be satisfied. (2).

(i) The multiplier rule :-

There exist constants λ_o and e_i together with a function

$J(t,X,\dot{X},\lambda) = \lambda_o f + \lambda_i(t)\phi_i \qquad \ldots 1.6.$

such that $J_{\dot{X}} = \int_{t_1}^{t_2} J_X \, dt + C_o$ ... 1.7.

and $\phi_i(t,X,\dot{X},0) = 0$

Where $\lambda_i(t)$ are the so called Lagrange Undetermined Multipliers. Here function J of equation 1.6 is the adjoined cost function formed by combining the original performance criterion function F of equation 1.1. with the differential constraints ϕ_i of equation 1.3. The subscripts $\dot{X}$ and X refer to partial derivatives of J with respect to these variables.

Equation 1.7 results on differentiation with respect to the independent variable time in the familiar form of the Euler Lagrange equations.

$$\frac{d}{dt}\left(\frac{\partial J}{\partial \dot{X}}\right) = \frac{\partial J}{\partial X} \qquad \text{... 1.8.}$$

(ii) The Transversatility Condition :-

$$\left((J-\dot{X}J_{\dot{X}})dt + J_{\dot{X}}dx\right)_{t_1}^{t_2} + \lambda_o dg + e_i d\rho_i = 0 \qquad \rho_i = 0 \qquad \text{... 1.9.}$$

These equations must hold at the ends of a trajectory independent of the choice of differentials dt_o, dx_o, dt_1 and dx_1.

The above two necessary conditions provide equations the solution of which produces the variables - U of equation 1.2 in the optimal control problem - which may minimise the cost function F of equation 1.1 . Other necessary conditions namely the Wereistrass Condition, the Clebsh Condition and the Second Variation of F may be considered as tests to determine if in fact the solution obtained does provide a minimum. (2).

It is the solution of the equations resulting from the application of the above necessary conditions for an optimum which is the problem referred to earlier. In general the equations to solve-namely the Euler Lagrange equation 1.8 together with the dynamic equation 1.3 subject to the boundary conditions 1.4 and those obtained from the Transversatility Condition 1.9 - result in a non-linear boundary value problem. There is no generalised method for the solution of such a problem.

References (3,4) indicate an approach based on the Newton-Raphson Operator - which under appropriate conditions provides an iterative solution to this problem. The following is an outline of this principle.

Newton-Raphson Solution of Non-Linear Boundary Value Problem :-

Consider Newton's method for the determination of the roots of a scalar function $f(x) = 0$. This consists of making an initial approximation of the root and determining a second and successive approximation as depicted in Figure 1.

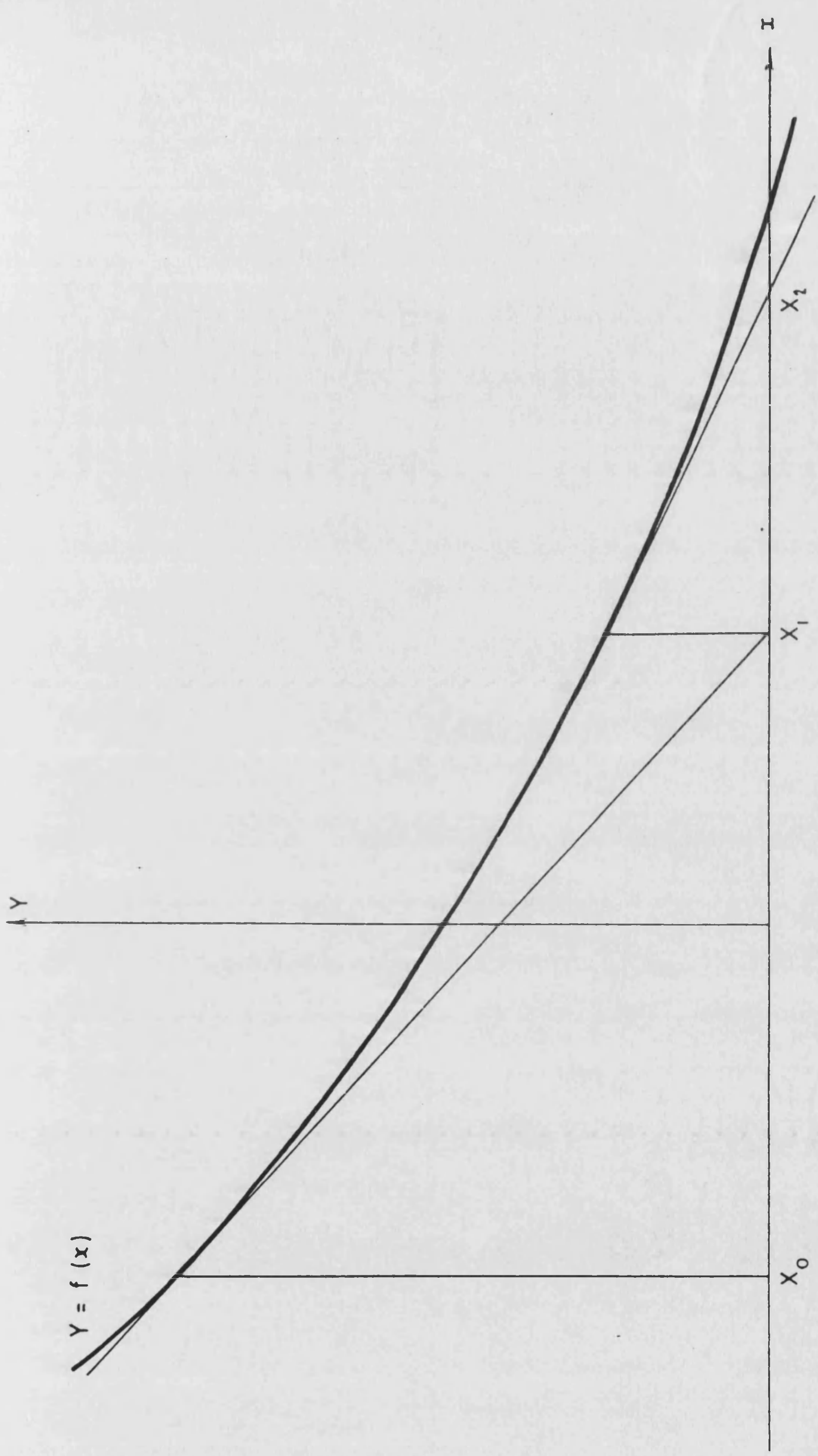
Taking the tangent to the curve $f(X) = 0$ at the point X_o leads to

$$f'(X_o) = f(X)/(X_o - X_1)$$

$$X_1 = X_o - f(X_o)/f'(X_o)$$

In general $X_{N+1} = X_N - f(X_N)/f'(X_N)$ or

$$f'(X_N) \times (X_{N+1} - X_N) + f(X_N) = 0$$


Y = f (x)
Y
x
X0
X1
X2

As can be seen from the figure the convergence is monotonic, i.e. successive approximations approach the true value from the initial approximation without oscillating about the true value. Another important feature - bearing in mind that for aircraft applications solutions must be obtained rapidly - is the fact that convergence is quadratic, indicating that few iterations need be performed.

This quadratic property is summarised as follows :

$$|X_{N+1} - X_N| \leqslant K|X_N - X_{N-1}|^2$$

The method generalises from the scalar example quoted to vector functional equations - the general algorithm for a non-linear set of differential equations of the form $(\dot{X}) = (G(X,t))$ being

$$(\dot{X}_{N+1}) = (J(X_N,t))\,(X_{N+1} - X_N) + (G(X_N,t)) \qquad \ldots\ 1.10.$$

where J is the Jacobian matrix of partial derivatives of $(G(X,t))$ with respect to (X).

The most significant point here is that although the original equations are non-linear, application of the Newton-Raphson algorthm reduces the non-linear boundary value problem to an iterative solution of linear differential equations. Matrix $(J(X_N,t))$ possesses time varying coefficients, however as these are known - having been obtained from the previous iterate - this does not detract from the method.

The principle of superposition may now be employed in the solution of the linearised equations to obtain a solution at each iterate which satisfies the boundary conditions. Each successive iteration will be a closer approximation to the desired vector which minimises the pay off.

Modification of unknown initial conditions.

The method employed to satisfy the boundary conditions at each iterate is as follows.

Consider a set of M-linear differential equations

$$(\dot{X}) = (A(t))\ (X) + (B(t))$$

with m of the M initial conditions unknown.

Generate a particular solution $Xp(t)$ of the equations, based on a set of starting vectors

$$X_N(t) \quad (i = 1, 2, \ldots . M)$$

which are the original approximations to the solution

$$X_{N+1}(t)\ (i = 1, 2, \ldots . M)$$

The initial conditions for this particular integration are made up of M-m known initial conditions and estimates for the m unknown initial conditions of $X_{N+1}(t_o)$.

In general the terminal conditions obtained from this solution will not satisfy the desired terminal conditions $X_D(t_f)$; as the estimates on the unknown initial conditions have been incorrect. In order that the desired terminal conditions $X_D(t_f)$ be met it is necessary to modify these original estimates. This is performed by generating sets of solutions $X^{j}_{iH}(t)$ $(i = j = (1,2, \ldots\ldots M))$ of the homogeneous system of equations $(\dot{X}) = (A(t))\ (X)$. The initial conditions for these homogeneous integrations being zero for $(M-1)$ of the variables and equal to 1 for the $X_{M-m+i}(t_o)$ variable. These homogeneous integrations are repeated for $i = 1,2, \ldots\ldots m$ and the terminal values $X^{j}_{iH}(t_f)$ noted. $(i=j=1,2,\ldots\ldots m)$. Because the differential equations are linear the following applies,

$$X^{j}_{D}(t_f) = X^{j}_{1}(t_f)C_1 + X^{j}_{2H}(t_f)C_2 + \ldots X^{j}_{iH}(t_f)C_i + X^{j}_{mH}(t_f)C_m + X^{j}_{P}(t_f) \quad \ldots 1.11.$$

$$(i = j = 1,2,\ldots\ldots m)$$

Since the magnitudes of $X^{j}_{D}(t_f)$; $X^{j}_{iH}(t_f)$ and $X^{j}_{P}(t_f)$ are known the constants C_i $(i=1,2,m)$ may be determined. These constants provide the modification to be made to the initial condition estimates of the m unknown initial conditions in order that the boundary conditions are satisfied. These modifications are

$$X^{j}_{N+1}(t_o) = X^{j}_{Estimate}(t_o) + C_j \qquad (j = 1,2, \ldots\ldots m) \quad \ldots 1.12.$$

In performing an integration now with the $(M-m)$ known initial conditions and the m modified initial conditions the solution X^{j}_{N+1} will be a better approximation to the nominal than X_N and the end conditions will be satisfied. The overall process is repeated until convergence occurs, using each successive solution $X^{j}_{N+1}(t)$ as the starting vector $X^{j}_{N}(t)$ for the next iteration.

SECTION 2.

AN APPLICATION OF THE THEORY.

Section 1 of this report has covered the relevant points of theory underlying the optimisation procedure to be employed in the identification phase of the adaptive control system. A Calculus of Variations approach has been used in order that the methods described may be extended using the Second Variation to the control law re-optimisation problem. The method of solution of the non-linear boundary value problem associated with the optimisation technique has been outlined.

To further illustrate the technique employed in the solution of boundary value problems consider the following example. Although a linear example this is worked in an identical manner to that for a non-linear system, to demonstrate the method.

Given the dynamics of a system

$$\dot{X} = -AX + A \qquad \ldots 2.1.$$

$$\dot{A} = 0$$

with boundary conditions $X_D(t_o) = 0$ and $X_D(t_f) = .9$ it is desired to find $X(t)$ $t < t < t_f$ and A such that the terminal conditions are satisfied at time $t_f = 1$ second.

Applying the generalised Newton-Raphson algorithm of equation 1.10 repeated here

$$\dot{X}_{N+1} = (J(X_N,t))\,(X_{N+1} - X_N) + (G(X_N,t))$$

to equations 2.1.

gives,

$$\begin{vmatrix} \dot{X}_{N+1} \\ \dot{A}_{N+1} \end{vmatrix} = \begin{vmatrix} -A_N & (-X_N + 1) \\ 0 & 0 \end{vmatrix} \begin{vmatrix} X_{N+1} - X_N \\ A_{N+1} - A_N \end{vmatrix} + \begin{vmatrix} -A_N X_N + A_N \\ 0 \end{vmatrix} \qquad \ldots 2.2.$$

which reduces to :-

$$\dot{X}_{N+1} = -A_N X_{N+1} + (1 - X_N) A_{N+1} + A_N X_N \qquad \ldots 2.3.$$

$$\dot{A}_{N+1} = 0$$

Starting Vectors for X_N and A_N are chosen. These are respectively,

$$X_N(t) = .5t \qquad 0 \leq t \leq 1 \qquad \ldots 2.4.$$

$$A_N(t) = .5$$

The initial condition on X_{N+1} is set to satisfy $X_D(t_o) = 0$ and an estimate $A_E(t_o)$ is made on the unknown initial condition.

In this example $A(t)$ is constant. In general all the variables would be time varying however no modification is required to the method.

A particular integration of 2.3 is carried out yielding $A_p(t_f)$ and $X_p(t_f)$ which does not in general satisfy $X_D(t_f)$ because the estimate for $A_E(t_o)$ is incorrect.

In order to modify the initial estimate a homogeneous integration is performed of the equations.

$$\dot{X}_{N+1} = -A_N X_{N+1} + (1-X_N)A_{N+1} \qquad \ldots\ 2.5.$$

$$\dot{A}_{N+1} = 0$$

With initial conditions $X_{N+1}(t_o) = 0$ and $A_{N+1}(t_o) = 1$

This produces $X_H(t_f)$

Making use of 1.11

$$X_D(t_f) = X_H(t_f)\ C_1 + X_p(t_f)$$

$$C_1 = \frac{X_D(t_f) - X_p(t_f)}{X_H(t_f)} \qquad \ldots\ 2.6.$$

Modifying initial conditions $A_{N+1}(t_o)$ as 1.12 gives

$$A_E^*(t_o) = A_E(t_o) + \frac{X_D(t_f)-X_p(t_f)}{X_H(t_f)}$$

An integration performed now with the same starting vector for $X_N(t)$ and using the modified estimate $A_E^*(t_o)$ produces $A_{N+1}(t)$; and $X_{N+1}(t)$ which satisfies the terminal conditions.

These vectors are now used as the new starting vectors and the process repeated until convergence occurs and no further iterations are required. A schematic representation of the steps involved is shown in figure 2.1. Convergence occurs when the difference between successive iterations is within some predetermined error level.

The results of a digital computation of this example are indicated in figure 2.2.

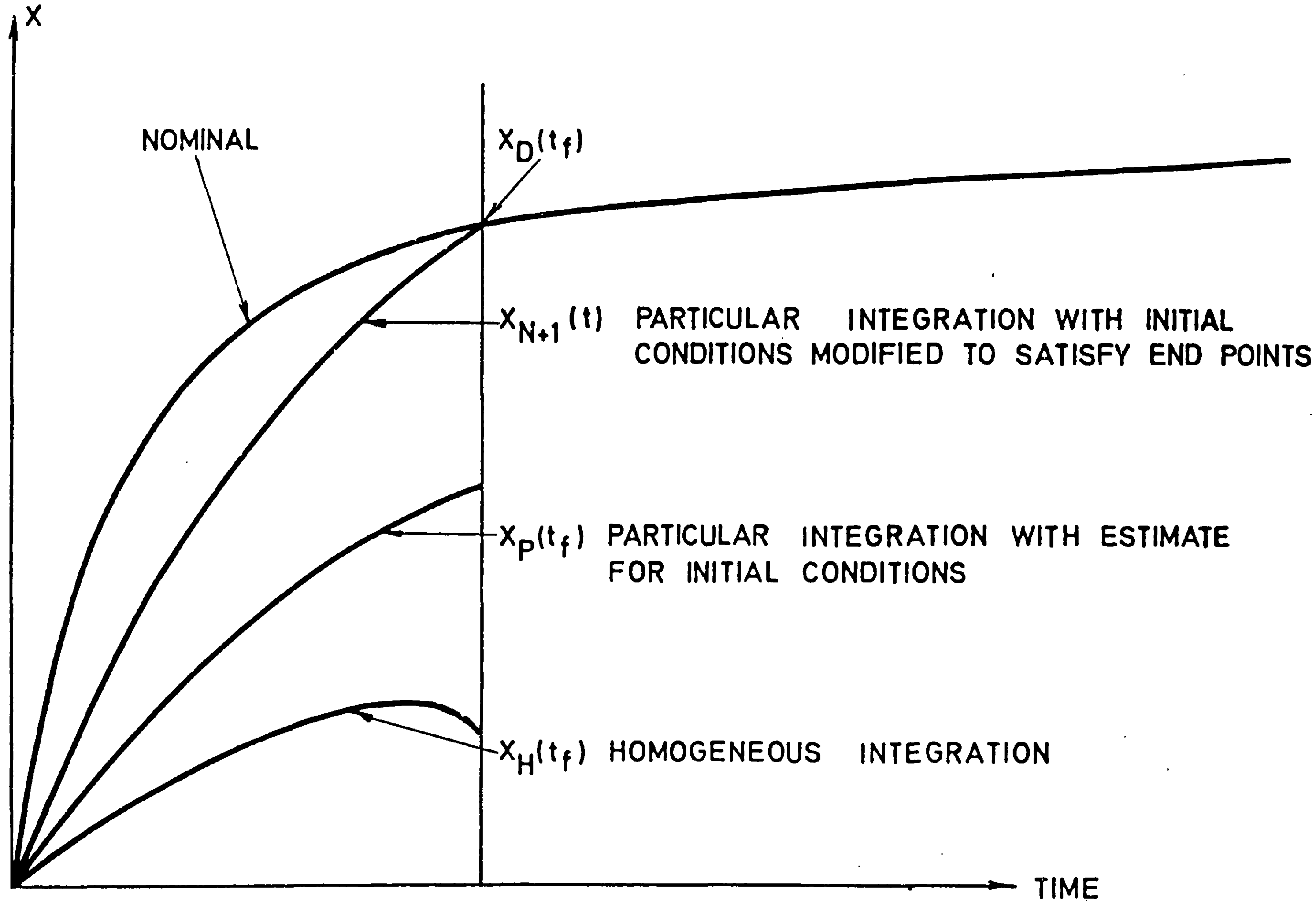

X
NOMINAL
$X_D(t_f)$
$X_{N+1}(t)$ PARTICULAR INTEGRATION WITH INITIAL CONDITIONS MODIFIED TO SATISFY END POINTS
$X_P(t_f)$ PARTICULAR INTEGRATION WITH ESTIMATE FOR INITIAL CONDITIONS
$X_H(t_f)$ HOMOGENEOUS INTEGRATION
TIME

FIG. 2: 1.

X(s) | λ(s) X(s) | λ(s)

0 6062 SEC.

STARTING VECTORS

0.00000000	0.50000000
0.05000000	0.50000000
0.10000000	0.50000000
0.15000000	0.50000000
0.20000000	0.50000000
0.25000000	0.50000000
0.30000000	0.50000000
0.35000000	0.50000000
0.40000000	0.50000000
0.45000000	0.50000000
0.50000000	0.50000000

FIRST ITERATE

0.00000000	1.30254610
0.11216407	[illegible]
0.25481256	1.30254610
0.36620048	1.30254610
0.46819721	1.30254610
0.56601670	[illegible]
0.64405711	1.30254610
0.71008550	1.30254610
0.78792662	1.30254610
0.84757450	1.30254610
0.90000000	1.30254610

SECOND ITERATE

0.00000000	2.11846107
0.15512587	2.11846107
0.33257596	2.11846107
0.46542089	2.11846107
0.58004145	2.11846107
0.67621619	2.11846107
0.74921523	2.11846107
0.79985611	2.11846107
0.84250497	2.11846107
0.87529827	2.11846107
0.90000000	2.11846107

THIRD ITERATE

0.00000000	2.30015718
0.20559674	2.30015718
0.35907700	2.30015718
0.45098545	2.30015718
0.50219771	2.30015718
0.63400710	2.30015718
0.74007774	2.30015718
0.80037275	2.30015718
0.84102767	2.30015718
0.87419128	2.30015718
0.90000000	2.30015718

FOURTH ITERATE

0.00000000	2.30258622
0.20557166	2.30258622
0.35904265	2.30258622
0.49861294	2.30258622
0.58169287	2.30258622
0.66577232	2.30258622
0.74681139	2.30258622
0.80047377	2.30258622
0.84151067	2.30258622
0.87410745	2.30258622
0.90000000	2.30258622

FIFTH ITERATE

0.00000000	2.30258510
0.20557177	2.30258510
0.35904266	2.30258510
0.49861277	2.30258510
0.58169283	2.30258510
0.66577223	2.30258510
0.74681130	2.30258510
0.80047377	2.30258510
0.84151068	2.30258510
0.87410746	2.30258510
0.90000000	2.30258510

This example of the solution of a two point boundary value problem indicates how the unknown parameter A may be determined such that the dynamics of the system satisfy specific terminal conditions. This is in fact an identification problem of a fixed parameter although no optimisation has been applied to the process. The problem however has been tackled in a manner which is applicable to the solution of boundary value problems arising out of an optimisation process using calculus of variations. If problems of this nature are to be solved by this iteration method then it is essential that solutions be obtained rapidly in order that on-line optimum identification may be carried out.

To demonstrate that it is indeed feasible to solve these two point boundary problems in a sufficiently short period the previous example was used with a time varying A where it was desired to track A continuously. Assuming A to be slowly time varying it is possible to assume that over short periods of time A is constant. By taking the initial and final values of the response of the system over this short period ($0 \leq t \leq t_1$) A may be determined using the above outlined iteration procedures. The technique is then repeated for the next short time interval ($t_1 \leq t \leq t_2$) etc. In this manner if the speed of solution is sufficiently rapid such that the time intervals are short, then the identified values of A as a function of time should closely approximate to the actual time varying A.

The simulation of this feasibility study is described in the next section.

SECTION 3.

HYBRID IMPLEMENTATION

On-Line Parameter Identification.

The iterative identification procedure of the previous example required an initial starting vector X_N. When convergence was complete the solution X_{N+1} satisfied the Terminal conditions. In a practical identification problem if a response has been monitored over a short period then the values at discrete points are in fact the true solution of the dynamics. It is therefore no longer necessary to estimate a starting vector for the state variables as the true values are available. An estimate on the initial condition of Λ_{N+1} is all that is now required. This results in fewer iterations being required to obtain a solution. As each iterate uses past values namely X_N and Λ_N, these must be stored. In view of this fact and since computation to determine new estimates for $\Lambda_{N+1}(t_o)$ have to be made, it is desirable to perform this part of the solution by making use of a digital computer.

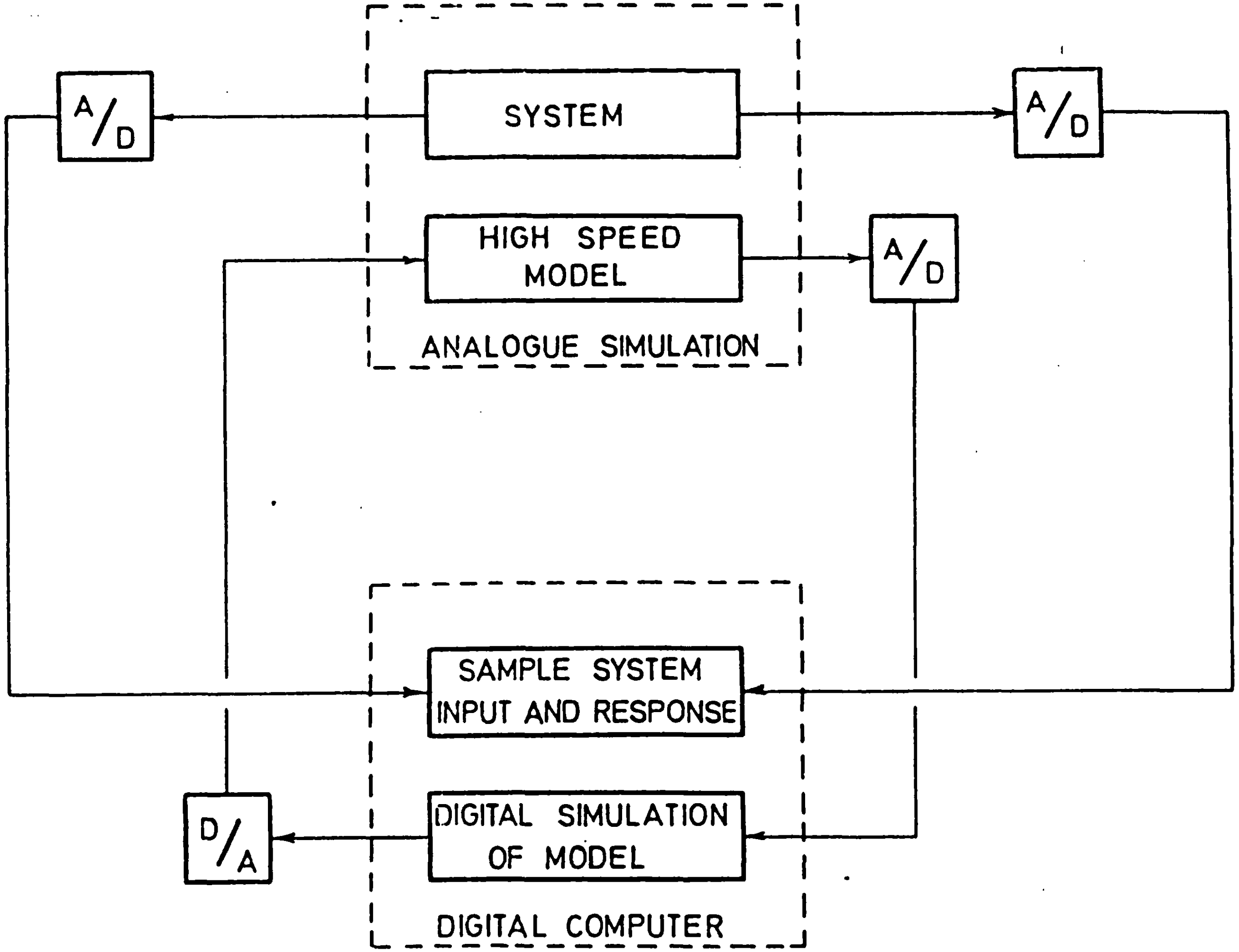
Each operation in a digital computer although fast in itself must be performed in series with other operations. Thus the accumulative time for numerical solution of a set of differential equations is relatively long compared with an analogue solution where multiple operations may be performed in parallel simultaneously. In order to achieve as short a time per iteration as possible it was decided to implement this feasibility study using Hybrid Computer techniques, making use of the analogue facilities for integrations, and digital facilities to control the state of the analogue, store state variable values and perform computation and logic operations required to test for convergence.

Since the sequence of operations for this on-line identification procedure is as follows,

1. Record State Variables and forcing function over a short time interval.
2. Use these variables as starting vectors for step 3.
3. Perform Particular Integral and Record($X_p t_f$).
4. Perform Homogeneous Integration.
5. Modify Estimate on $A_{N+1}(t_o)$
6. Iterate steps 3 through 5 until convergence test is satisfied.
7. Repeat procedure from step 1 for next time interval.

it is essential that the identification time required between record phases be as short as possible. It is thus desirable to perform the integration steps 3 and 4 on a faster than real time scale, enabling the total number of iterations for convergence to be performed in a very short real time.

The analogue simulation of equations 2.3 for the purposes of performing faster than real time solutions of the particular and homogeneous integrations may be considered to be a high speed parameter tracking model. A block diagram showing the implementation lay-out is provided in figure 3.1.


A/D
SYSTEM
A/D
HIGH SPEED MODEL
A/D
ANALOGUE SIMULATION
SAMPLE SYSTEM INPUT AND RESPONSE
D/A
DIGITAL SIMULATION OF MODEL
DIGITAL COMPUTER

The initial simulation was not concerned with speed of solution but more with overcoming hybrid implementation problems and accuracy of solution. Some of these problems are now described.

To perform floating point calculations requires use of either Floating Point Package - which is merely a slight extension of the basic machine language for the P-D-P-8 digital computer - or the Fortran Operating system. Although speed of operation is reduced by using Fortran it is possible to keep track of the programme more easily and therefore Fortran Language was used. It is necessary to leave the main Fortran Programme in order to perform Analogue/Digital and Digital/Analogue conversions, this is accomplished using a PAUSE statement in the main programme.

Interface conversions are performed to 12-bit accuracy which is equivalent to 1 location of storage space. Floating Point storage makes use of 3-locations per number. It is therefore desirable to store interface conversions as integer arrays to economise in storage space. In order to preserve accuracy by reducing round off error, calculations are performed in floating point mode by first converting integers to this form and then reconverting to integer mode at the end of a calculation. These operations are a compromise between accuracy, speed of computation and storage space.

It is necessary to scale variables for Digital/Analogue conversion such that integer values do not exceed the maximum of $\pm 2^{11}$ this being performed in floating point by multiplying by a scaling factor before conversion to integer mode.

The first simulation ran both model and system on the same time scale. Sampling of the system and outputing forcing functions to the model in the same DO LOOP to ensure synchronisation between model and system.

This simulation was demonstrated in detail to Messrs. Watts and Shanks of the Royal Aircraft Establishment, Farnborough on their visit to Swansea University during March 1967.

Results obtained for this hybrid simulation were in close agreement with digital studies. The second simulation using a high speed tracking model was then implemented.

For this on-line simulation an additional problem was synchronisation between model and system as these are now operating on different time scales. In order to achieve this synchronisation the programme executed a model integration and at the end of this sampled a timing signal from the analogue computer. The magnitude of the signal level representing this extremely short time interval was then multiplied by a time scaling factor to give a signal level corresponding to the real time equivalent of the model integration period.

If the number of samples to be taken of the system response and forcing function during this time interval is N and the model integration time is T then each sample of system must occur at intervals of $\frac{T}{(N-1)}$ x (THE SCALING FACTOR).

A practical point of interest here is that it is not sufficient to evaluate the above expression as an integer - it is required in integer form for comparison with time signals sampled from system - and merely increment the value obtained by itself for each successive interval. If this is done then any round off error in the integer conversion is accumulated.

It is necessary to determine sampling time at each interval in floating point form and then convert to integer mode for comparison with system timing signal in order to preserve timing accuracy and synchronisation.

A flow diagram of the digital computer programme is depicted in figure 3.2. The programme listing is given in the Appendix.

Traces of Results obtained from this simulation are shown in figure 3.3 through figure 3.5.

Figures 3.3 a, b and c show the identification of three different fixed parameters. The process is running slowly to indicate the various steps involved. Model traces show particular integrations followed by homogeneous integrations. The starting estimate of A is the same in each case, namely .5 . The model is running on a 2:1 time scale with respect to the system and the response of the system is due to a unit step forcing function. The identification process is not continuous and after convergence the process halts. The test for convergence is a difference between

A_{N+1} and A_N of less than one percent.

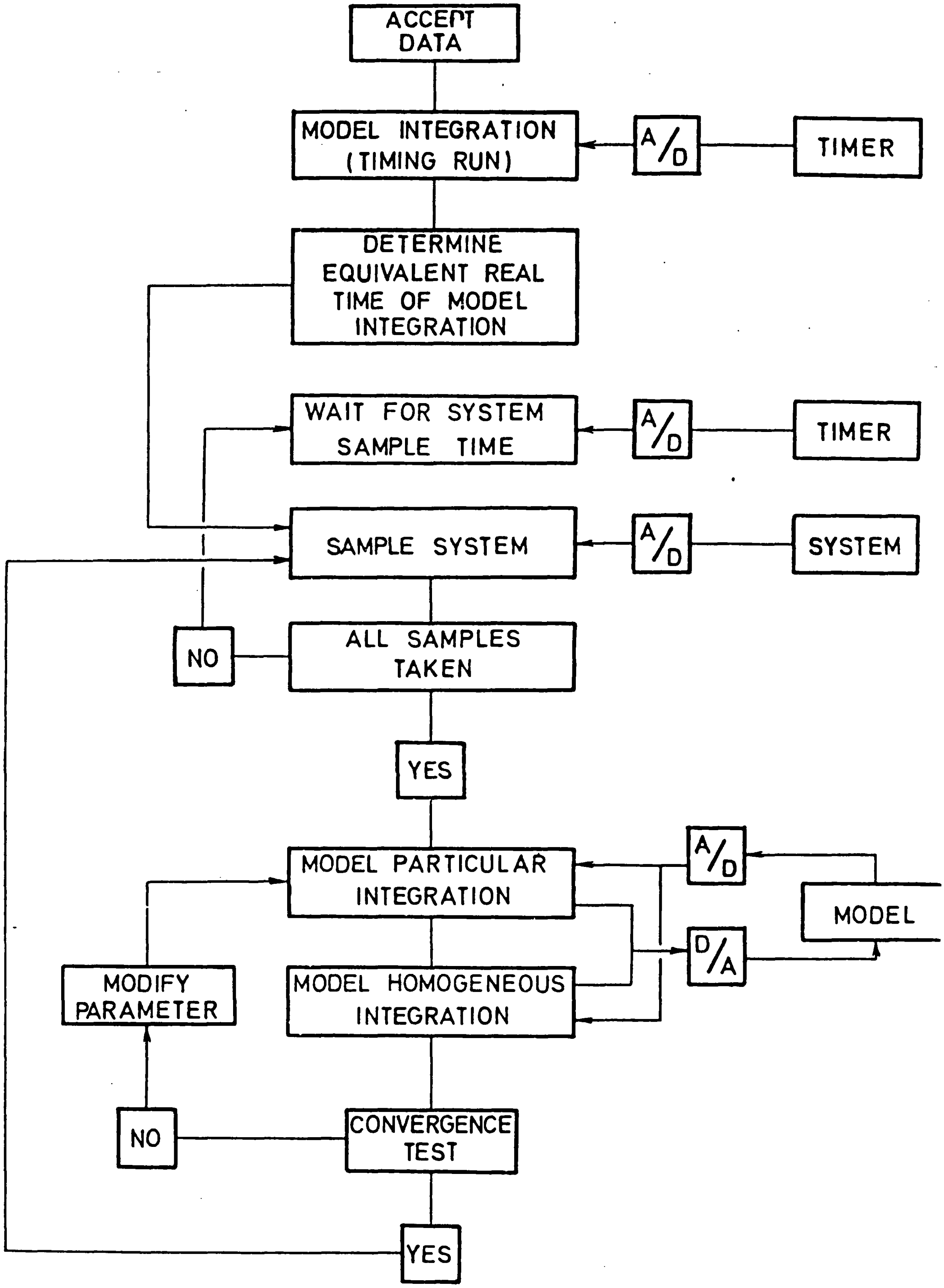
Figure 3.4(a) is a portion of a continuous identification of a fixed parameter the system being repeatedly perturbed by a triangular waveform. Convergence test and model time scaling are as for figure 3.3. Note that model particular integration traces have zero initial conditions. Initial conditions are stored in the digital computer obviating the need to reset them on the analogue computer integrators for successive identification.

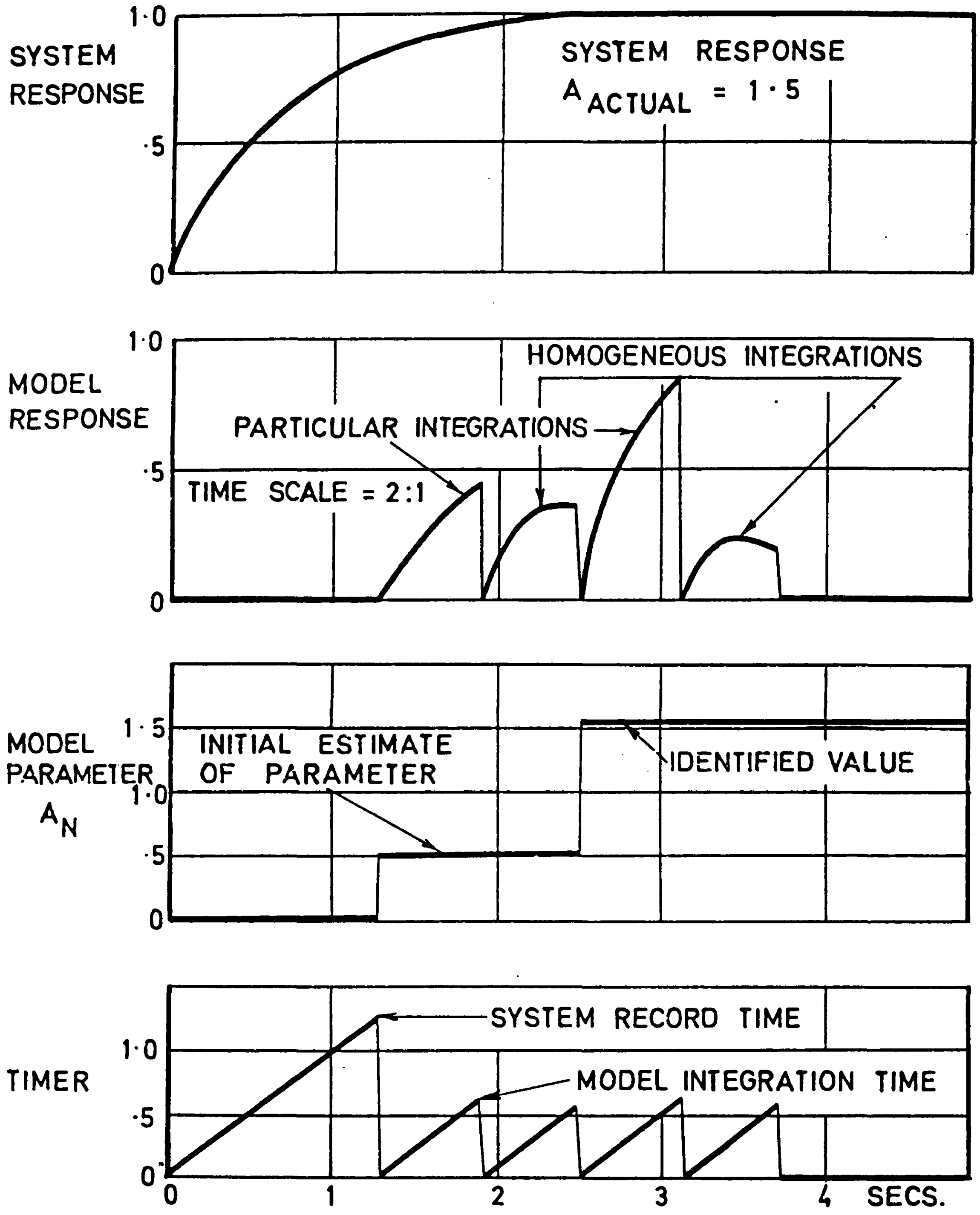
Figure 3.4(b) is a continuous identification of parameter the system having been excited by a step input. It is important to note that as the system approaches steady state conditions the magnitude of the homogeneous integration tends to zero. Expression 2.6 then becomes indeterminate. This is to be expected as in the steady state configuration there is no unique solution for A_{N+1} which can be seen by A_N diverging from the true value. As the convergence test is no longer satisfied the system is not resampled for the next identification the process trying in vain to satisfy the convergence test.

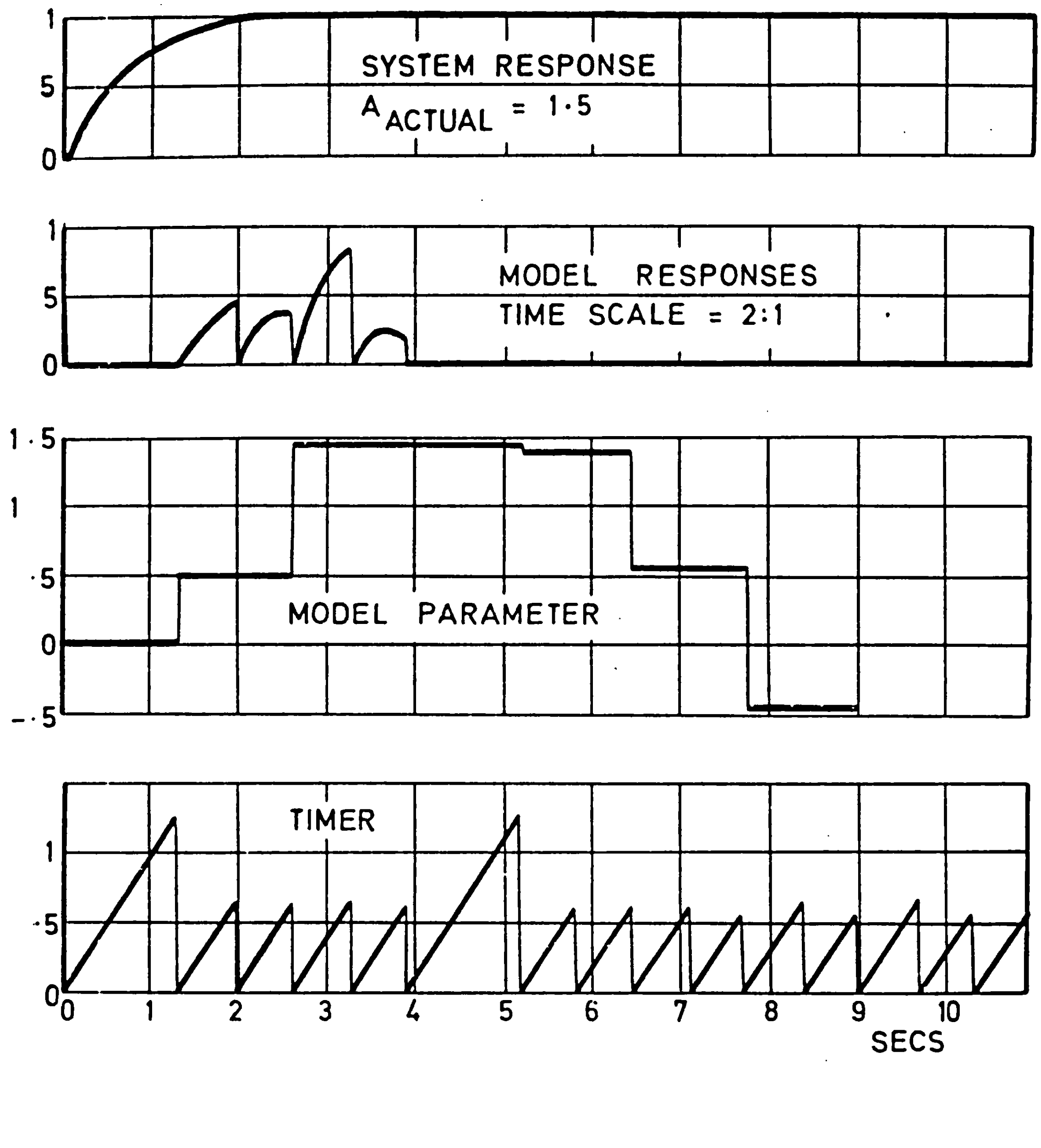
To overcome this failing of the implementation the convergence test was modified. Convergence was assumed to occur when the error between system response and model particular integral was less than 1%. If this state was satisfied then the homogeneous integration associated with this particular integral was not performed and the programme recorded a new sample of system response without modifying the current state of the identified value of parameter.

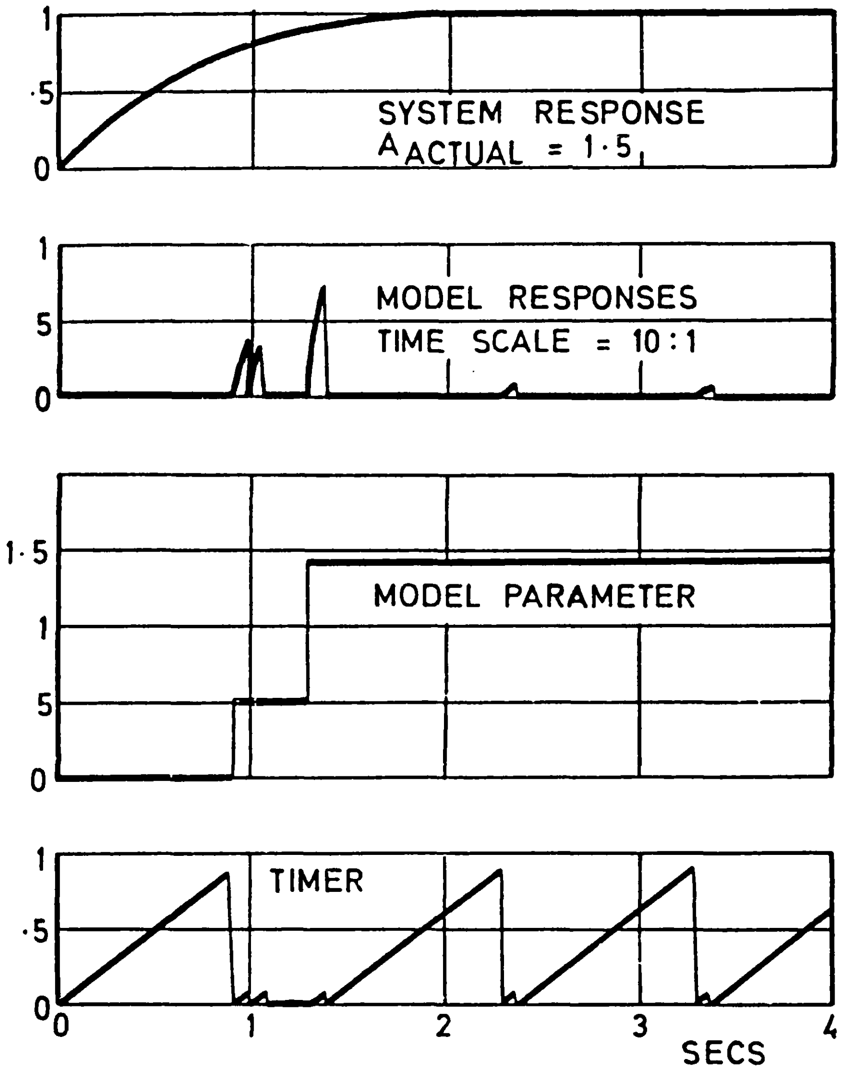
Figure 3.5(a) indicates the continuous identification process using the above mentioned alteration to the convergence test. Model to system time scaling is 10:1. The failing of figure 3.4(b) has been eliminated. All of the previous results have been performed slowly.

Figure 3.5(b) indicates a faster identification the record time being .45 seconds approximately. The convergence test is .1% error between system and model, this being the reason for more than one iteration before identification is complete. Figure 3.6 shows the response of the system to a triangular waveform. The parameter is being continuously varied sinusoidly over the range .5 to 3 a change of 6:1 . Record time is .1 second and the model is on a 12:1 time scale with respect to the system.


ACCEPT DATA
MODEL INTEGRATION (TIMING RUN)
A/D
TIMER
DETERMINE EQUIVALENT REAL TIME OF MODEL INTEGRATION
WAIT FOR SYSTEM SAMPLE TIME
A/D
TIMER
SAMPLE SYSTEM
A/D
SYSTEM
NO
ALL SAMPLES TAKEN
YES
MODEL PARTICULAR INTEGRATION
A/D
MODEL
D/A
MODIFY PARAMETER
MODEL HOMOGENEOUS INTEGRATION
NO
CONVERGENCE TEST
YES

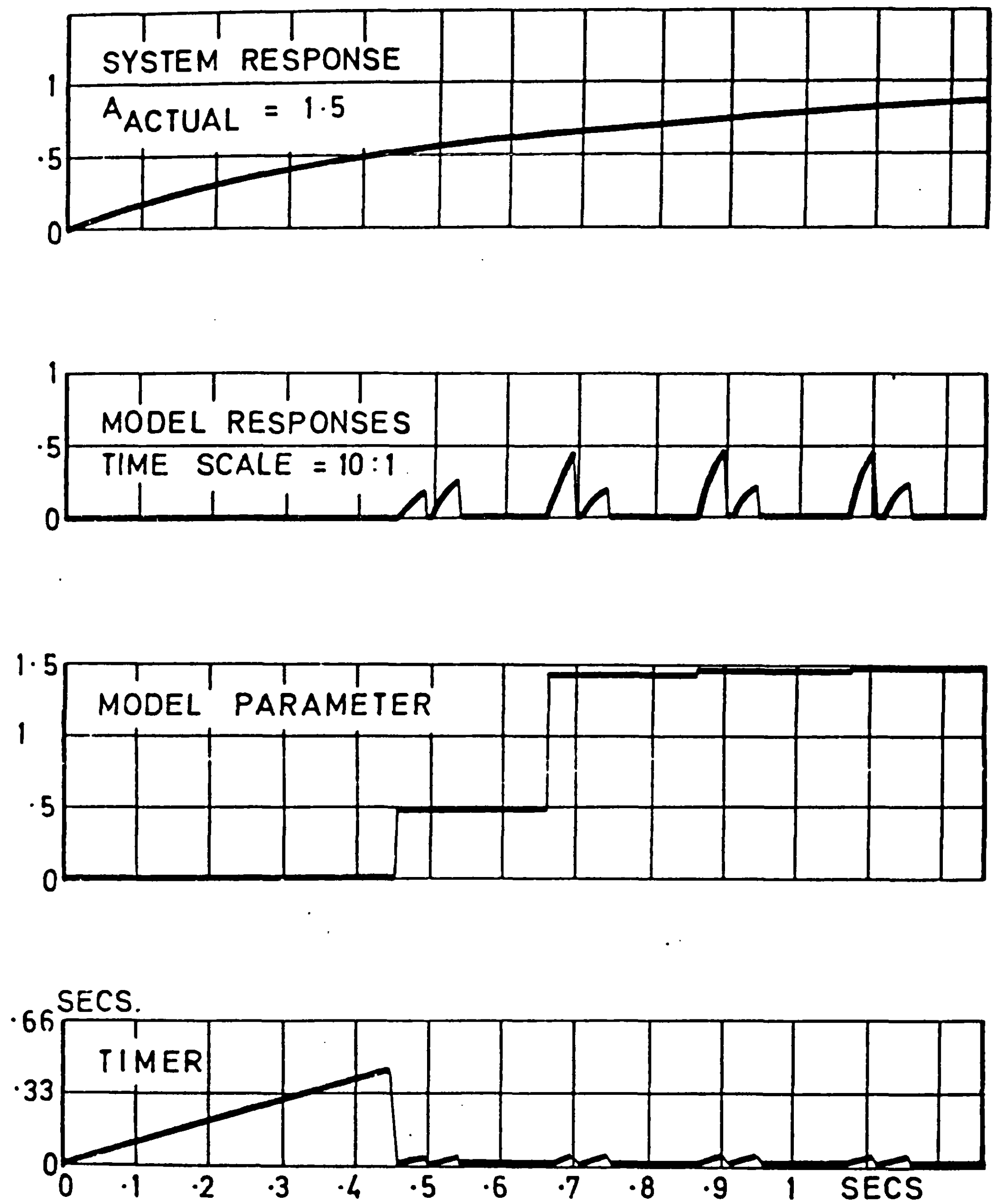

1·0
SYSTEM
RESPONSE
·5
0
SYSTEM RESPONSE
A ACTUAL = 1·5
1·0
MODEL
RESPONSE
·5
0
HOMOGENEOUS INTEGRATIONS
PARTICULAR INTEGRATIONS
TIME SCALE = 2:1
MODEL
PARAMETER
A N
1·5
1·0
·5
0
INITIAL ESTIMATE
OF PARAMETER
IDENTIFIED VALUE
TIMER
1·0
·5
0
SYSTEM RECORD TIME
MODEL INTEGRATION TIME
0
1
2
3
4
SECS.


SYSTEM RESPONSE
A ACTUAL = 1·5
MODEL RESPONSES
TIME SCALE = 2:1
MODEL PARAMETER
TIMER
SECS
0
1
2
3
4
5
6
7
8
9
10
1
5
0
1·5
·5
-·5



CONTINUOUS IDENTIFICATION

FIG. 3:5a.


SYSTEM RESPONSE
A ACTUAL = 1·5
1
·5
0
MODEL RESPONSES
TIME SCALE = 10:1
1
·5
0
MODEL PARAMETER
1·5
1
·5
0
SECS.
TIMER
·66
·33
0
0 ·1 ·2 ·3 ·4 ·5 ·6 ·7 ·8 ·9 1 SECS

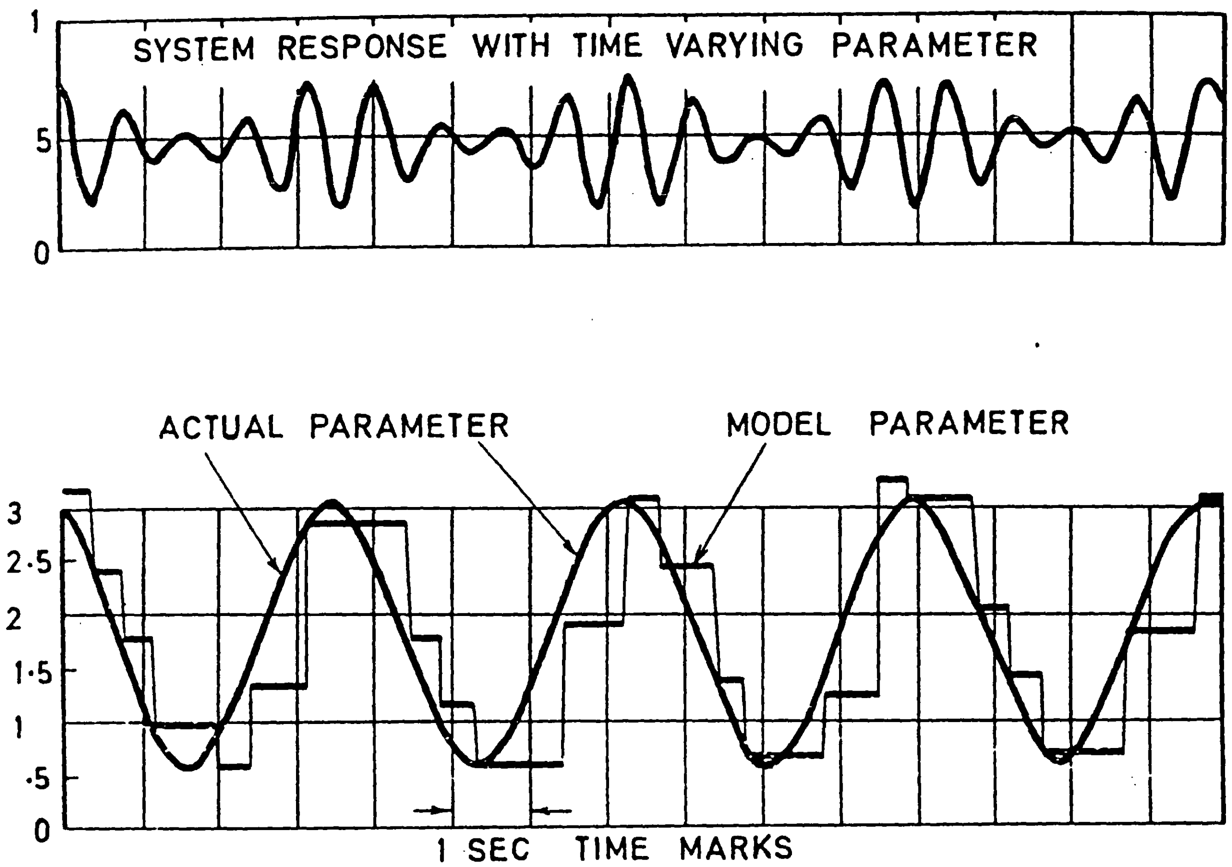
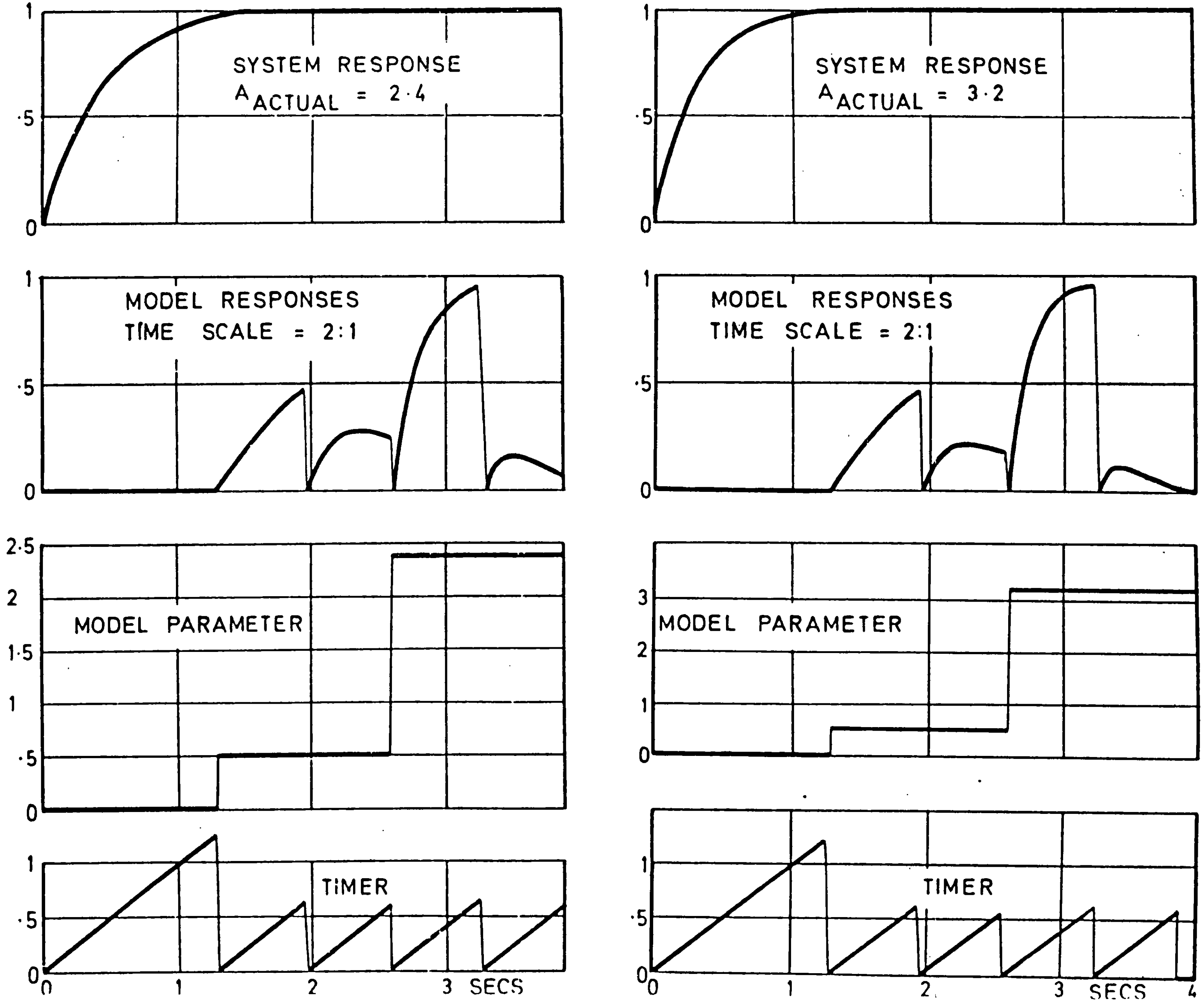
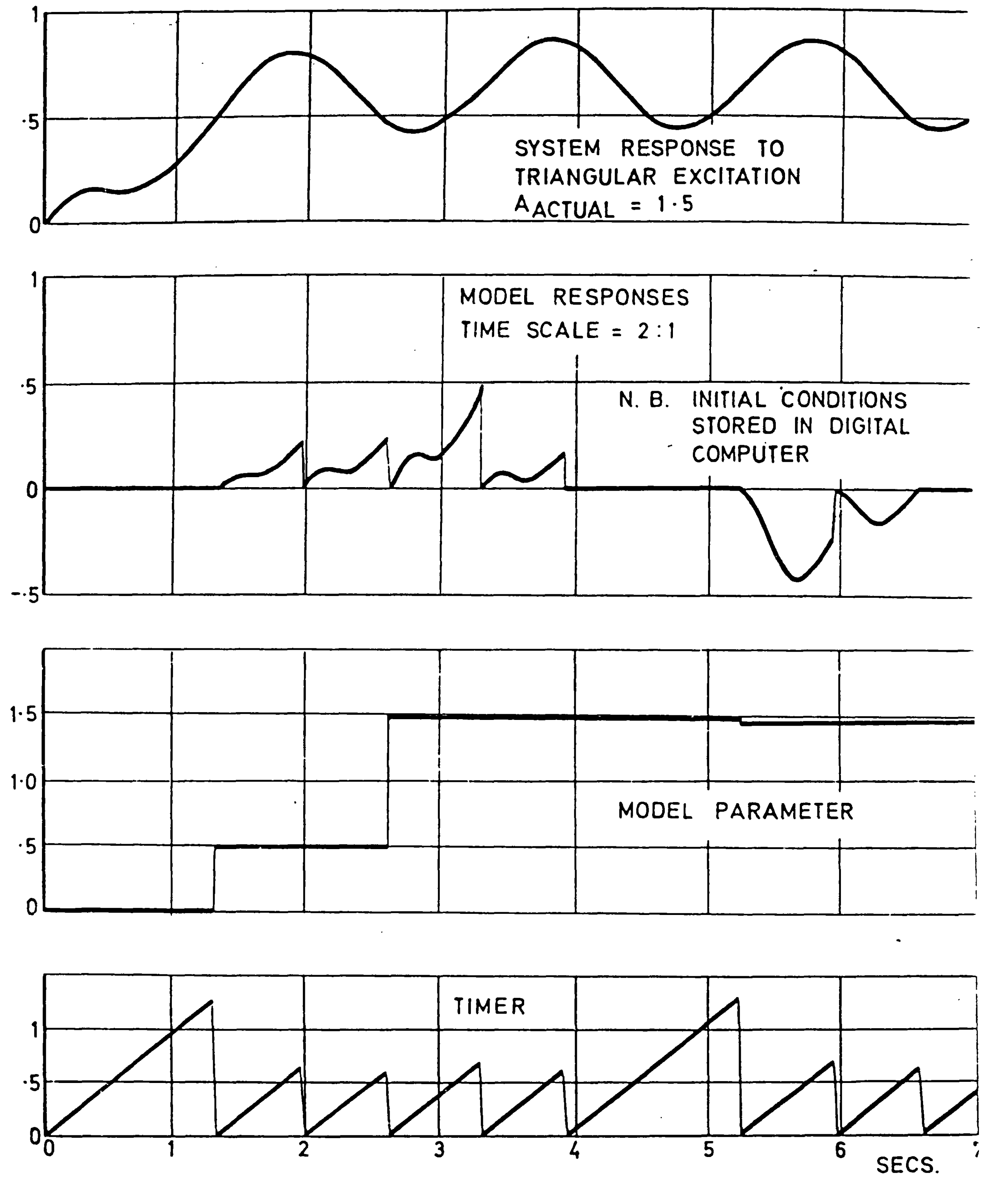

SYSTEM RESPONSE WITH TIME VARYING PARAMETER
1
5
0
ACTUAL PARAMETER
MODEL PARAMETER
3
2·5
2
1·5
1
·5
0
1 SEC TIME MARKS

FIG 3:6


SYSTEM RESPONSE
A ACTUAL = 2·4
MODEL RESPONSES
TIME SCALE = 2:1
MODEL PARAMETER
TIMER
SECS
SYSTEM RESPONSE
A ACTUAL = 3·2
MODEL RESPONSES
TIME SCALE = 2:1
MODEL PARAMETER
TIMER
SECS



SYSTEM RESPONSE TO TRIANGULAR EXCITATION
A_{ACTUAL} = 1·5
MODEL RESPONSES
TIME SCALE = 2 : 1
N. B. INITIAL CONDITIONS STORED IN DIGITAL COMPUTER
MODEL PARAMETER
TIMER
SECS.

SECTION 4.

INFERENCE OF RESULTS.

An On-Line identification of a time varying parameter has been performed. The example chosen was of necessity simple - due to the availability of only two digital to analogue converters - however the approach to the solution has been made in a manner applicable to non-linear systems of higher order. The accuracy and speed of solution have shown the technique to be of value for incorporation in an adaptive flight control system. Multiparameter identification would require more homogeneous integrations to be performed per iteration. It would be possible however to execute these simultaneously as indeed the homogeneous and particular integration of the example could have been with consequently no additional time penalty. Questions relating to the convergence properties of the generalised Newton-Raphson Method remain unanswered. Convergence is dependent on how close the starting vector approximations are to the true solution. Reference (3). By making use of the true solution of the dynamics as starting vectors convergence properties are enhanced.

The primary reason for implementation of this hybrid solution was as a feasibility study for the solution of boundary value problems arising out of optimisation techniques. It has been shown that this iterative solution is sufficiently fast to enable the method to be applied to the solution of these problems and thus the next stage of development is to incorporate optimisation in the identification process. This optimum identification will have a pay off criterion which is a function of time varying parameters enabling them to be identified without the assumption that they are piecewise constant.

A penalty for noisy measurements of system response will also be included to assist in overcoming problems associated with turbulence. This is to be the subject of the author's next report. It is also intended to consider problems concerning the implementation of the re-optimisation phase of the adaptive control process.

Multiplexing equipment for the Digital to Analogue converters is at present under construction. When this is completed it will be possible to extend the study to systems which are more relevant.

REFERENCES

1. OPTIMISATION AND CONTROL OF NON-LINEAR SYSTEMS USING THE SECOND VARIATION.
   John V. Breakwell; Jason E.L. Speyer and Arthur E. Bryson.
   J.S.I.A.M. Control. 1963.

2. LECTURES ON THE CALCULUS OF VARIATIONS.
   G.A. Bliss. The University of Chicago Press, Chicago 1946.

3. ON NON-LINEAR DIFFERENTIAL EQUATION, THE MAXIMUM OPERATION AND MONOTONE CONVERGENCE.
   R. Kalaba. Journal Math.Mech.8,519(1959).

4. QUASILINEARISATION AND NON-LINEAR BOUNDARY VALUE PROBLEMS.
   Richard E. Bellman and Robert E. Kalaba.
   American Elsevier Publishing Company, Inc. 1965.

5. VARIATIONAL METHOD IN PROBLEMS OF CONTROL AND PROGRAMMING.
   Leonard D. Berkovitz. Journal of Mathematical Analysis and Applications, p.145-169, 1961.

Appendix B

STEEPEST-DESCENT OPTIMISATION PROGRAMME

```
      DIMENSION TCD(12),X(3,12),A(3,12),Y(13,12)
      DIMENSION E(3,12),C(3,12,12)
      DIMENSION ICUS(12)
      OPEN (4,FILE='OUT.DAT',STATUS='UNKNOWN',ACCESS='SEQUENTIAL',
     1 FORM='FORMATTED')
      OPEN(12,FILE='OUT1.DAT',STATUS='UNKNOWN',ACCESS='DIRECT',
     1 FORM='UNFORMATTED',RECL=1000)
      OPEN(7,FILE='OUT2.DAT',STATUS='UNKNOWN',ACCESS='DIRECT',
     1 FORM='UNFORMATTED',RECL=500)
      CALL DAUX10(N,NV,NU,NUH,NT,NG,NIS,DT,ITT,NH,TCD,X,ICUS)
      STOP
      END


      SUBROUTINE DAUX10(N,NV,NU,NUH,NT,NG,NIS,DT,ITT,NH,TCD,X,ICUS)
      DIMENSION X(3,12),TCD(12),C(12,12),E(12),Y(13,12)
      DIMENSION H(12,12),B(12),R(12),FIC(12),G(12),P(12),X0(12)
      DIMENSION ICUS(12),U(101)
20    FORMAT(9E14.4)
      TAU=.000000002
      NU=7
      LLL=0
      LLLL=0
      NUH=12
      IR=1
      N=10
      NV=11
      NH=5
      DT=.83*4.
      ITT=10
      NG=10
      NIS=10
      NT=NG*NIS
      IRNW=1
      K=1
      IT=0
      CALL INIT(X0)
      DO 5 I=1,5
    5 X(1,I)=X0(I)
      X(1,11)=X0(11)
      U0=X0(11)
      CALL WPI(N,NV,NU,X,K,IT,IRNW)
      DO 1 I=1,NT
      CALL FINT(X0,N,DT,ITAG,U0)
      CALL A11(X0,U0)
   10 WRITE(4,*)I,(X0(J),J=1,5),X0(11)
```

```
      DO 6 J=1,5
    6 X(1,J)=X0(J)
      X(1,11)=X0(11)
      IT=I
         CALL WPI(N,NV,NU,X,K,IT,IRNW)
1        CONTINUE
         CALL WYOT(N,NV,NU,NT,1)
      RETURN
      DO 50 LL=1,75
      IF(LLLL-LL)90,91,90
   90 CALL TERM(X0)
      IRNW=2
      DO 7 J=1,NV
    7 X(1,J)=X0(J)
      CALL FNC(C,E,X0,U0,UNEW,TAU)
      U(101)=UNEW
      IT=100
      CALL WPI(N,NV,NU,X,K,IT,IRNW)
      WRITE(4,*)IT,(X0(J),J=1,NV)
      IRNR=199
      ITAG=0
      DO 8 I=NT-1,0,-1
      CALL BINT(X0,N,DT,ITAG,U0,UNEW,TAU)
      CALL RPI(N,NV,NU,X,K,IT,IRNR)
      IRNR=IRNR-4
      DO 9 J=1,5
    9 X0(J)=X(1,J)
      X0(11)=X(1,11)
      WRITE(4,*)I,(X0(J),J=1,NV)
      CALL FNC(C,E,X0,U0,UNEW,TAU)
      U(I+1)=UNEW
      DO 15 J=1,NV
   15 X(1,J)=X0(J)
      IT=I
      CALL WPI(N,NV,NU,X,K,IT,IRNW)
    8 CONTINUE
      WRITE(4,*)(I,U(I),I=1,NT+1)
      IRNW=1
      I=0
      IT=0
      TEST=X0(11)-U(1)
      IF(ABS(TEST)-.00001)70,70,71
   70 LLL=LLL+1
      WRITE (4,*)'CONVERGENCE',LLL
      TAU=TAU*2.
      IF(LLL-1)71,80,72
   80 TAU=TAU*2.
   72 IF(LLL-3)71,73,71
   73 LLLL=LL+2
   71 WRITE(4,*)I,(X0(J),J=1,5),X0(11)
      CALL WPI(N,NV,NU,X,K,IT,IRNW)
      ITAG=0
      DO 11 I=1,NT
      CALL FINT(X0,N,DT,ITAG,U0)
```

```
      X0(11)=U(I+1)
      WRITE(4,*)I,(X0(J),J=1,5),X0(11)
      DO 16 J=1,5
   16 X(1,J)=X0(J)
      X(1,11)=X0(11)
      IT=I
         CALL WPI(N,NV,NU,X,K,IT,IRNW)
11       CONTINUE
50    CONTINUE
91    IRNR=202
      IRNW=1
      DO 55 I=1,NT+1
      CALL RPI(N,NV,NU,X,K,IT,IRNR)
      IRNR=IRNR-4
55       CALL WPI(N,NV,NU,X,K,IT,IRNW)
         RETURN
         END

     SUBROUTINE INIT(Y)
     DIMENSION Y(12)
     CALL IC(Y,N,DT,NT)
     U0=Y(11)
     I=0
     WRITE(4,*)I,(Y(J),J=1,N),U0
     RETURN
     END

     SUBROUTINE TERM(Y)
     DIMENSION Y(12)
     CALL TC(Y,N,DT,NT)
     U0=Y(11)
     I=100
     WRITE(4,*)I,(Y(J),J=1,N),U0
     RETURN
     END

     SUBROUTINE IC(X,N,DT,NT)
     DIMENSION X(12)
     N=5
     NT=100
     DT=3.32
     X(1)=400.
     X(2)=0.0
     X(3)=700.
     X(4)=0.0
     X(5)=42000./32.2
     READ(5,*)X(11)
     RETURN
     END
     SUBROUTINE TC(A,N,DT,NT)
     DIMENSION A(12),X(3,12)
     N=10
     NV=11
```

```
      NU=7
      NT=100
      IRNR=201
      K=1
      CALL RPI(N,NV,NU,X,K,IT,IRNR)
      DO 1 I=1,NV
    1 A(I)=X(1,I)
      A(6)=A(1)-968.586
      A(7)=0.0
      A(8)=-1.
      A(9)=0.0
      A(10)=0.0
      RETURN
      END
      SUBROUTINE FNC(C,E,X,U0,UNEW,TAU)
      DIMENSION C(12,12),E(12),X(12)
      GRAV=32.2
      CGRAV=GRAV*1600.0
      SG=SIN(X(2))
      CG=COS(X(2))
      GCG=GRAV*CG
      GSG=GRAV*SG
      CA=COS(X(11))
      SA=SIN(X(11))
      H1=27300.
      S=530.
      CALL ROE1(X(3),ROE)
      CALL MACH(X(1),X(3),FMACH,A)
      CALL THRUST(FMACH,X(3),T,DTM,DTH)
      ROEVS=ROE*X(1)*S
      QS=0.5*ROEVS*X(1)
      CALL AERO(FMACH,CLA,CD0,CLN,DCLDM,DCLNM,DCDM)
      DLDADV=ROEVS*(CLA+0.5*FMACH*DCLDM)
      DTDV=DTM/A
      DTDH=DTH
      DTDVDV=0.0
      DTDVDH=-DTDV/H1
      DTDHDH=-DTH/H1
      DLDA=QS*CLA
      DLDADA=0.0
      DDDADA=2.*QS*CLN
      DLDADH=-DLDA/H1
      DDDA=2.*QS*CLN*X(11)
      DDDADH=-DDDA/H1
      VDDADV=2.*QS*X(11)*(2.*CLN+FMACH*DCLNM)
      DDDADV=VDDADV/X(1)
      DLDV=DLDADV*X(11)
      DLDVDH=-DLDV/H1
      DLDVDV=(1.0/X(1))*(DLDV+1.5*ROEVS*FMACH*DCLDM*X(11))
      FL=QS*CLA*X(11)
      DLDH=-FL/H1
      DLDHDH=FL/(H1*H1)
      D=QS*(CD0+CLN*X(11)*X(11))
```

```
      DDDH=-D/H1
      DDDHDH=D/(H1*H1)
      DDDV=(2.*D/X(1))+.5*ROEVS*FMACH*(DCDM+DCLNM*X(11)*X(11))
      DDDVDH=-DDDV/H1
      DDDVDV=(1.0/X(1))*(DDDV+1.5*ROEVS*FMACH*(DCDM
     1+DCLNM*X(11)*X(11)))
      E(1)=-GSG+(T*CA-D)/X(5)
      E(2)=(-GCG+(FL+T*SA)/X(5))/X(1)
      E(3)=X(1)*SG
      E(4)=X(1)*CG
      E(5)=-T/CGRAV
      E(6)=(-1./X(5))*(X(6)*(CA*DTDV-DDDV)+X(7)*(DLDV+DTDV*SA-E(2)
     1*X(5))/X(1))-X(8)*SG-X(9)*CG+X(10)*DTDV/CGRAV
      E(7)=GCG*X(6)-(GSG*X(7)/X(1))-X(8)*X(1)*CG
     1+X(9)*X(1)*SG
      E(8)=((-X(6)*(DTDH*CA-DDDH)-X(7)*(DLDH+DTDH*SA)/X(1))/
     1X(5))+X(10)*DTDH/CGRAV
      E(9)=0.0
      E(10)=(-X(6)*(D-T*CA)+X(7)*(FL+T*SA)/X(1))/(X(5)*
     1X(5))
      UNEW=X(11)-TAU*(X(11)+(-X(6)*(T*SA+DDDA)+X(7)
     1*(DLDA+T*CA)/X(1))/X(5))
      RETURN
      END
      SUBROUTINE DK(E,C,S,D,K,DT,N)
      DIMENSION E(12),C(12,12),S(12),D(4,12)
      DO 10 I=1,N
      TEMP=E(I)
C       DO 11 J=1,N
C   11 TEMP=TEMP+C(I,J)*S(I)
   10 D(K,I)=DT*TEMP
      RETURN
      END
      SUBROUTINE FINT(A,N,DT,ITAG,U0)
      DIMENSION A(12),S(12),D(4,12),C(12,12),E(12)
C   20 DO 22 I=1,N
C       DO 22 J=1,N
C   22 C(I,J)=0.0
   21 DO 10 I=1,N
   10 S(I)=A(I)
      S(11)=A(11)
      CALL FNC(C,E,S,U0,UNEW,TAU)
      CALL DK(E,C,S,D,1,DT,N)
      DO 11 I=1,N
   11 S(I)=A(I)+.5*D(1,I)
      CALL FNC(C,E,S,U0,UNEW,TAU)
      CALL DK(E,C,S,D,2,DT,N)
      DO 12 I=1,N
   12 S(I)=A(I)+.5*D(2,I)
      CALL FNC(C,E,S,U0,UNEW,TAU)
      CALL DK(E,C,S,D,3,DT,N)
      DO 13 I=1,N
   13 S(I)=A(I)+D(3,I)
```

```
      CALL FNC(C,E,S,U0,UNEW,TAU)
      CALL DK(E,C,S,D,4,DT,N)
      DO 14 I=1,N
   14 A(I)=A(I)+(D(1,I)+2.*D(2,I)+2.*D(3,I)+D(4,I))/6.
      RETURN
      END

      SUBROUTINE A11(A,U0)
      DIMENSION A(12)
      A(11)=U0*((400./A(1))**2.)
      IF(A(11)-.03)50,51,51
   50 A(11)=.03
   51 CONTINUE
      RETURN
      END

      SUBROUTINE BINT(A,N,DT,ITAG,U0,UNEW,TAU)
      DIMENSION A(12),S(12),D(4,12),C(12,12),E(12)
      DT=-DT
      IF(ITAG)20,20,21
   20 DO 22 I=1,N
      DO 22 J=1,N
   22 C(I,J)=0.0
      ITAG=1
   21 DO 10 I=1,N
   10 S(I)=A(I)
      S(11)=A(11)
      CALL FNC(C,E,S,U0,UNEW,TAU)
      CALL DK(E,C,S,D,1,DT,N)
      DO 11 I=1,N
   11 S(I)=A(I)+.5*D(1,I)
      CALL FNC(C,E,S,U0,UNEW,TAU)
      CALL DK(E,C,S,D,2,DT,N)
      DO 12 I=1,N
   12 S(I)=A(I)+.5*D(2,I)
      CALL FNC(C,E,S,U0,UNEW,TAU)
      CALL DK(E,C,S,D,3,DT,N)
      DO 13 I=1,N
   13 S(I)=A(I)+D(3,I)
      CALL FNC(C,E,S,U0,UNEW,TAU)
      CALL DK(E,C,S,D,4,DT,N)
      DO 14 I=1,N
   14 A(I)=A(I)+(D(1,I)+2.*D(2,I)+2.*D(3,I)+D(4,I))/6.
      DT=-DT
      RETURN
      END

      SUBROUTINE THRUST(FMACH,AH,T,DTDM,DTDH)
      DIMENSION RH(11),RM(10)
      OPEN(10,FILE='LOG2.THM',STATUS='UNKNOWN',ACCESS='SEQUENTIAL',
     1 FORM='FORMATTED')
    4 FORMAT(F10.4,2F10.4)
```

```
C     WRITE(6,*)'MACH=',FMACH,'HT=',AH
      FMACHE=FMACH
      FH=AH/1000.
      IF(FH)60,61,61
60    FH=0.001
61    IF(FMACH)62,63,63
62    FMACH=0.0
63    RH(1)=0.
      DO 10 I=2,7
   10 RH(I)=RH(I-1)+5.
      RH(8)=40.
      RH(9)=50.
      RH(10)=70.
      RH(11)=80.
C     FIND NEAREST HEIGHT
      DO 11 I=1,11
      IF(FH-RH(I))12,11,11
   11 CONTINUE
      I=I-1
   12 DH1=RH(I)-RH(I-1)
      DH=FH-RH(I-1)
      FRACH=DH/DH1
      HR1=RH(I-1)
      HR2=RH(I)
C     SET REF. MACH.
      RM(1)=0.
      DO 31 J=2,10
   31 RM(J)=RM(J-1)+.2
      DO 13 J=1,10
      IF (FMACH-RM(J))30,13,13
   13 CONTINUE
      J=J-1
   30 DM=FMACH-RM(J-1)
      FRACM=DM/.2
C     WRITE(6,*)'FRACM=',FRACM
      RM1=RM(J-1)
      RM2=RM(J)
      REWIND 10
   15 READ(10,4,END=55)T1,H1,FM1
      TEST=ABS(RM1-FM1)
      IF (TEST-.001)14,14,15
   14 TEST=ABS(HR1-H1)
      IF(TEST-1.)16,16,17
   17 READ(10,4,END=55)T1,H1,FM1
      GO TO 14
   16 READ(10,4,END=55)T2,H2,FM1
      DTM1=T2-T1
   18 READ(10,4,END=55)T3,H1,FM2
      TEST=ABS(RM2-FM2)
      IF (TEST-.001)19,19,18
   19 TEST=ABS(HR1-H1)
      IF(TEST-1.)20,20,21
   21 READ(10,4,END=55)T3,H1,FM2
      GO TO 19
```

```
   20 READ(10,4,END=55)T4,H2,FM2
      DTM2=T4-T3
      TCM1=T1+FRACH*DTM1
      TCM2=T3+FRACH*DTM2
      T=(TCM1+FRACM*(TCM2-TCM1))*1000.
      DTDM=((TCM1-TCM2)/(FM1-FM2))*1000.
      TCH1=T1+FRACM*(T3-T1)
      TCH2=T2+FRACM*(T4-T2)
      DTDH=((TCH1-TCH2)/(H1-H2))
C     WRITE(6,*)'T=',T,'MACH=',FMACH
      FMACH=FMACHE
      RETURN
   55 WRITE(6,*) 'READ ERROR UNIT 10'
      STOP
      END
      SUBROUTINE ROE1(HF,ROE)
      ROE0=.00254
      H1=27300.
      IF(HF)10,11,11
10    HFT=0.0
      GO TO 12
11    HFT=HF
12    Y=2.7182818**(-HFT/H1)
      ROE=ROE0*Y
      RETURN
      END
      SUBROUTINE MACH(V,HF,FM,A)
      IF(HF)10,11,11
10    HFT=0.0
      GO TO 12
11    HFT=HF
12    HREF=11.0*1000.0*100.0/(2.54*12.)
      IF (HFT-HREF)2,2,3
3     H1=HREF
      GO TO 4
2     H1=HFT
4     A=(33146.0/(2.54*12.))*SQRT((288.15-0.00198*H1)/273.15)
      FM=V/A
      RETURN
      END

      SUBROUTINE ALPHA2(J,X,Y,IACON)
      DIMENSION X(3,12),Y(13,12)
      C=X(J,11)
      H1=27300.
      S=530.
      DM=Y(1,5)-X(J,5)
      DV=Y(1,1)-X(J,1)
      DH=Y(1,3)-X(J,3)
      DLV=Y(1,6)-X(J,6)
      DLG=Y(1,7)-X(J,7)
      WRITE(6,*)'DV=',DV,'DH=',DH,'DLV=',DLV,'DLG=',DLG
C     PAUSE
```

```
      CALL ROE1(X(J,3),ROE)
      CALL MACH(X(J,1),X(J,3),FMACH,A)
      CALL THRUST(FMACH,X(J,3),T,DTM,DTH)
      ROEVS=ROE*X(J,1)*S
      QS=0.5*ROEVS*X(J,1)
      CALL AERO(FMACH,CLA,CD0,CLN,DCLDM,DCLNM,DCDM)
      DLDADV=ROEVS*(CLA+0.5*FMACH*DCLDM)
      DTDV=DTM/A
      DTDVDV=0.0
      DTDVDH=-DTDV/H1
      DTDHDH=-DTH/H1
      DLDA=QS*CLA
      DLDADA=0.0
      DDDADA=2.*QS*CLN
      DO 1 L=1,1
C     DO 1 L=1,2
      M=1
      IF(L-1)4,4,5
    5 M=800
    4 CONTINUE
      DO 1 I=1,M
      IF(L-1)6,6,7
    7 CONTINUE
      IF(ABS(Y(1,11)-C)-.00001)2,2,3
    3 CONTINUE
      C=Y(1,11)
    6 CONTINUE
      WRITE(6,*)'C=',C
      DLDADH=-DLDA/H1
      DDDA=2.*QS*CLN*C
      DDDADH=-DDDA/H1
      VDDADV=2.*QS*C*(2.*CLN+FMACH*DCLNM)
      DDDADV=VDDADV/X(J,1)
      SA=SIN(C)
      CA=COS(C)
      DG1DA=1.0+(-X(J,6)*(T*CA+DDDADA)+X(J,7)*(DLDADA-T*SA)
     1/X(J,1))/X(J,5)
      Z=-DG1DA*25.
      DG1DV=(-X(J,6)*(DTDV*SA+DDDADV)+(X(J,7)/(X(J,1)*X(J,1)))
     1*(X(J,1)*(DLDADV+CA*DTDV)-(DLDA+T*CA)))/X(J,5)
      DG1DH=(-X(J,6)*(SA*DTH+DDDADH)+(X(J,7)/X(J,1))*
     1(DLDADH+CA*DTH))/X(J,5)
      DG1DLV=-(T*SA+DDDA)/X(J,5)
      DG1DLG=(DLDA+T*CA)/(X(J,5)*X(J,1))
      G1=X(J,11)+(-X(J,6)*(T*SA+DDDA)+X(J,7)*(DLDA+T*CA)/X(J,1))/X(J,5)
      DG1DM=(X(J,11)-G1)/X(J,5)
      E1=-(X(J,1)*DG1DV+X(J,3)*DG1DH-G1)
      Y(1,11)=C+(DG1DV*DV+DG1DH*DH+DG1DLV*DLV
     1+DG1DLG*DLG+DG1DM*DM+G1)/Z
      GO TO 80
   80 WRITE(6,*)'Y11=',Y(1,11),'Z=',Z,'G1DV=',DG1DV,'G1DH=',DG1DH
    1 CONTINUE
      IACON=1
      GO TO 40
```

```
 2 CONTINUE
   IACON=0
40 WRITE(6,*)C
   DLDV=DLDADV*C
   DLDVDH=-DLDV/H1
   DLDVDV=(1.0/X(J,1))*(DLDV+ROEVS*FMACH*DCLDM*C)
   FL=QS*CLA*C
   DLDH=-FL/H1
   DLDHDH=FL/(H1*H1)
   D=QS*(CD0+CLN*C*C)
   DDDH=-D/H1
   DDDHDH=D/(H1*H1)
   DDDV=(2.*D/X(J,1))+.5*ROEVS*FMACH*(DCDM+DCLNM*C*C)
   DDDVDH=-DDDV/H1
   DDDVDV=(1.0/X(J,1))*(DDDV+ROEVS*FMACH*(DCDM+DCLNM*C*C))
   WRITE(6,*)'FL=',FL,'DRAG=',D,'ALPHA=',C
C    PAUSE
   RETURN
   END
   SUBROUTINE AERO(FMACH,CLA,CD0,CLN,DCLAM,DCLNM,DCDM)
   DIMENSION RM(10)
   OPEN(11,FILE='LOG.CLD',STATUS='UNKNOWN',ACCESS='SEQUENTIAL',
  1 FORM='FORMATTED')
 4 FORMAT(F10.4,4F10.4)
   FMACHE=FMACH
   IF(FMACHE)2,3,3
 2 FMACH=0.0
 3 RM(1)=.0
   RM(2)=.4
   RM(3)=.8
   RM(4)=.9
   RM(5)=1.0
   DO 10 I=6,10
10 RM(I)=RM(I-1)+.2
   DO 11 I=1,9
   IF(FMACH-RM(I))12,11,11
11 CONTINUE
   I=I-1
12 DM=FMACH-RM(I-1)
   DM1=RM(I)-RM(I-1)
   FM1=RM(I-1)
   FM2=RM(I)
   REWIND 11
14 READ (11,4,END=20)FM,CL1,CD1,FN1,CLN1
   TEST=ABS(FM1-FM)
   IF (TEST-.01)13,13,14
13 READ (11,4,END=20)FM,CL2,CD2,FN2,CLN2
   DCLAM=(CL2-CL1)/DM1
   DCDM=(CD2-CD1)/DM1
   DCLNM=(CLN2-CLN1)/DM1
   CLA=CL1+DCLAM*DM
   CD0=CD1+DCDM*DM
   CLN=CLN1+DCLNM*DM
```

```
C    WRITE(6,*)'CLA=',CLA,'CD0=',CD0,'CLN=',CLN
C    WRITE(6,*)'DCLAM=',DCLAM,'DCDM=',DCDM,'DCLNM=',DCLNM
     FMACH=FMACHE
     RETURN
  20 WRITE(6,*)'READ ERROR UNIT 11'
     STOP
     END

     SUBROUTINE WYTIN(A,NU,IR,IRN,IT,N,NV,NH)
     DIMENSION A(13,12)
     J=NH+1
     WRITE(NU,REC=IR)N,J,IT,((A(K,I),I=1,N),K=1,J)
     IRN=IR+1
     RETURN
     END
     SUBROUTINE RYTIN(A,NU,IR,IRN,IT,N,NV,NH)
     DIMENSION A(13,12)
     J=NH+1
     READ(NU,REC=IR)N,J,IT,((A(K,I),I=1,N),K=1,J)
     IRN=IR+1
     RETURN
     END
        SUBROUTINE WYOT(N,NV,NU,NT,M)
        DIMENSION A(3,12)
10      FORMAT(1I4,9E14.4)
        L=1
        IX=1
        IY=NV
        IF(N-9)1,1,2
2       IY=9
        L=2
1       K=1
        DO 3 J=1,L
        GO TO (4,5),J
5       IX=10
        IY=NV
4       IRN=M
        DO 6 I=1,NT+1
        CALL RPI(N,NV,NU,A,K,IT1,IRN)
        WRITE (4,10)IT1,(A(1,IJ),IJ=IX,IY)
6       CONTINUE
3       CONTINUE
        RETURN
        END
        SUBROUTINE WPI(N,NV,NU,A,K,IT,IRNW)
        DIMENSION A(3,12)
        IRC=IRNW
        WRITE(NU,REC=IRC)IT,(A(K,I),I=1,NV)
        IRNW=IRNW+2
        RETURN
        END
        SUBROUTINE RPI(N,NV,NU,A,K,IT,IRNR)
        DIMENSION A(3,12)
```

```
      IRC=IRNR
      READ(NU,REC=IRC)IT,(A(K,I),I=1,NV)
      IRNR=IRNR+2
      RETURN
      END


      SUBROUTINE ZCE(N,C,E)
      DIMENSION C(3,12,12),E(3,12)
      DO 1 I=1,3
      DO 1 J=1,N
      E(I,J)=0.
      DO 1 K=1,N
      C(I,J,K)=0.
1     CONTINUE
      RETURN
      END
```

Appendix C.

QUASILINEARISATION OPTIMISATION PROGRAMME.

```
      DIMENSION TCD(12),X(3,12),A(3,12),Y(13,12)
         DIMENSION E(3,12),C(3,12,12)
         DIMENSION ICUS(12)
      OPEN (4,FILE='OUT.DAT',STATUS='UNKNOWN',ACCESS='SEQUENTIAL',
     1 FORM='FORMATTED')
         OPEN(12,FILE='OUT1.DAT',STATUS='UNKNOWN',ACCESS='DIRECT',
     1 FORM='UNFORMATTED',RECL=1000)
         OPEN(7,FILE='OUT2.DAT',STATUS='UNKNOWN',ACCESS='DIRECT',
     1 FORM='UNFORMATTED',RECL=500)
         NU=7
         NUH=12
         IR=1
         N=10
         NV=11
         NH=5
         DT=.83*4.
         ITT=10
         NG=10
         NIS=10
         NT=NG*NIS
         IRNW=1
         K=1
      IT=0
      ITAG=0
         CALL WYOT(N,NV,NU,NT,1)
      TCD(1)=968.586
      TCD(2)=0.
         TCD(3)=-1.
      TCD(4)=0.
         TCD(5)=0.
         ICUS(1)=1
      ICUS(2)=7
      ICUS(3)=8
      ICUS(4)=9
      ICUS(5)=10
         CALL ZCE(N,C,E)
         IFC=0
         L=1
         M=2
      ITAG3=1
         DO 3 K=1,20
      READ(5,*)IP
C     IP=1
         IRITS=1
         CALL DAUX2(N,NV,NU,NUH,Y,L,X,IRITS,IRN,NH,ICUS,ITAG3)
C     WRITE(6,*)(X(1,J),J=1,NV)
```

```
C     WRITE(6,*)((Y(I,J),J=1,NV),I=1,NH+1)
        ITAG3=1
     ITAG=1
        IRNR=L+2
        DO 11 II=1,NG
        DO 1 I=1,NIS
C     WRITE(6,*)(X(1,IJ),IJ=1,NV)
        CALL DAUX3(N,NV,NU,IRNR,ITAG,C,E,X,K1,IP)
C     GO TO (75,76)IP
C76   DO 78 KKK=1,3
C     DO 78 JJJ=1,N
C78    WRITE(4,*)'C=',(C(KKK,JJJ,III),III=1,N)
75      CALL INTRNG(Y,C,E,N,NH,DT)
        IRT=IRN
        IT=I
        CALL WYTIN(Y,NUH,IRT,IRN,IT,N,NV,NH)
     WRITE(4,*)IT,N,(Y(1,IJ),IJ=1,N),X(K1,11)
1       CONTINUE
        IRL=IRT
        CALL GSO(N,NV,NH,NUH,IRL,IRITS,Y)
11      CONTINUE
        CALL MODIC(N,NV,NH,IRL,IRITS,ICUS,NUH,TCD,Y)
        IT=0
     WRITE(4,*)'IT=',IT,(Y(1,JJ),JJ=1,N)
        KK=1
        IRNR=L
        CALL RPI(N,NV,NU,X,KK,IT,IRNR)
        CALL ALPHA2(KK,X,Y,IACON)
        DO 4 I=1,NV
        A(1,I)=Y(1,I)
4       CONTINUE
        IRNW=M
        CALL WPI(N,NV,NU,A,1,IT,IRNW)
        NH1=0
        ITAG=1
        DO 2 I=1,NT
C     WRITE(6,*)'I=',I
        CALL DAUX3(N,NV,NU,IRNR,ITAG,C,E,X,K1,IP)
        CALL INTRNG(Y,C,E,N,NH1,DT)
        J=K1
     CALL ALPHA2(J,X,Y,IACON)

C     IF(IACON)40,40,41
C 40  U=Y(1,11)
C     GO TO 42
C 41  Y(1,11)=U
C 42  WRITE(4,*)I,(Y(1,J),J=1,NV)

42      DO 5 J=1,NV
        A(1,J)=Y(1,J)
5       CONTINUE
        IT=I
        CALL WPI(N,NV,NU,A,1,IT,IRNW)
     CALL MACH(Y(1,1),Y(1,3),FMACH,SS)
```

```
      WRITE(4,*)I,FMACH,(Y(1,J),J=1,NV)
2         CONTINUE
          CALL CONVT(N,NV,NU,NT,ICON)
C         CALL WYOT(N,NV,NU,NT,M)
      IF(ICON-1)6,7,6
6         CONTINUE
          I=L
          L=M
          M=I
3         CONTINUE
7         STOP
          END
      SUBROUTINE THRUST(FMACH,AH,T,DTDM,DTDH)
      DIMENSION RH(11),RM(10)
      OPEN(10,FILE='LOG2.THM',STATUS='UNKNOWN',ACCESS='SEQUENTIAL',
     1 FORM='FORMATTED')
    4 FORMAT(F10.4,2F10.4)
C     WRITE(6,*)'MACH=',FMACH,'HT=',AH
      FMACHE=FMACH
      FH=AH/1000.
      IF(FH)60,61,61
60    FH=0.001
61    IF(FMACH)62,63,63
62    FMACH=0.0
63     RH(1)=0.
      DO 10 I=2,7
   10 RH(I)=RH(I-1)+5.
      RH(8)=40.
      RH(9)=50.
      RH(10)=70.
      RH(11)=80.
C     FIND NEAREST HEIGHT
      DO 11 I=1,11
      IF(FH-RH(I))12,11,11
   11 CONTINUE
      I=I-1
   12 DH1=RH(I)-RH(I-1)
      DH=FH-RH(I-1)
      FRACH=DH/DH1
      HR1=RH(I-1)
      HR2=RH(I)
C     SET REF. MACH.
      RM(1)=0.
      DO 31 J=2,10
   31 RM(J)=RM(J-1)+.2
      DO 13 J=1,10
      IF (FMACH-RM(J))30,13,13
   13 CONTINUE
      J=J-1
   30 DM=FMACH-RM(J-1)
      FRACM=DM/.2
C     WRITE(6,*)'FRACM=',FRACM
      RM1=RM(J-1)
      RM2=RM(J)
```

```
      REWIND 10
   15 READ(10,4,END=55)T1,H1,FM1
      TEST=ABS(RM1-FM1)
      IF (TEST-.001)14,14,15
   14 TEST=ABS(HR1-H1)
      IF(TEST-1.)16,16,17
   17 READ(10,4,END=55)T1,H1,FM1
      GO TO 14
   16 READ(10,4,END=55)T2,H2,FM1
      DTM1=T2-T1
   18 READ(10,4,END=55)T3,H1,FM2
      TEST=ABS(RM2-FM2)
      IF (TEST-.001)19,19,18
   19 TEST=ABS(HR1-H1)
      IF(TEST-1.)20,20,21
   21 READ(10,4,END=55)T3,H1,FM2
      GO TO 19
   20 READ(10,4,END=55)T4,H2,FM2
      DTM2=T4-T3
      TCM1=T1+FRACH*DTM1
      TCM2=T3+FRACH*DTM2
      T=(TCM1+FRACM*(TCM2-TCM1))*1000.
      DTDM=((TCM1-TCM2)/(FM1-FM2))*1000.
      TCH1=T1+FRACM*(T3-T1)
      TCH2=T2+FRACM*(T4-T2)
      DTDH=((TCH1-TCH2)/(H1-H2))
C     WRITE(6,*)'T=',T,'MACH=',FMACH
      FMACH=FMACHE
      RETURN
   55 WRITE(6,*) 'READ ERROR UNIT 10'
      STOP
      END
      SUBROUTINE ROE1(HF,ROE)
      ROE0=.00254
      H1=27300.
      IF(HF)10,11,11
10    HFT=0.0
      GO TO 12
11    HFT=HF
12    Y=2.7182818**(-HFT/H1)
      ROE=ROE0*Y
      RETURN
      END
      SUBROUTINE MACH(V,HF,FM,A)
      IF(HF)10,11,11
10    HFT=0.0
      GO TO 12
11    HFT=HF
12    HREF=11.0*1000.0*100.0/(2.54*12)
      IF (HFT-HREF)2,2,3
3     H1=HREF
      GO TO 4
2     H1=HFT
4     A=(33146.0/(2.54*12))*SQRT((288.15-0.00198*H1)/273.15)
```

```
 FM=V/A
 RETURN
 END

 SUBROUTINE DAUX3(N,NV,NU,IRNR,ITAG,C,E,X,K1,IP)
 DIMENSION C(3,12,12),E(3,12),X(3,12)
 L=1
 IF(ITAG-1)2,1,2
2 CONTINUE
 DO 3 I=1,N
 E(1,I)=E(3,I)
 DO 3 J=1,N
 C(1,I,J)=C(3,I,J)
3 CONTINUE
 L=2
 GO TO 4
1 CONTINUE
 K=3
 K1=1
 I=1
 J=1
7 CONTINUE
 GRAV=32.2
 CGRAV=GRAV*1600.0
 SG=SIN(X(J,2))
 CG=COS(X(J,2))
 GCG=GRAV*CG
 GSG=GRAV*SG
 CA=COS(X(J,11))
 SA=SIN(X(J,11))
 H1=27300.
 S=530.
 CALL ROE1(X(J,3),ROE)
 CALL MACH(X(J,1),X(J,3),FMACH,A)
 CALL THRUST(FMACH,X(J,3),T,DTM,DTH)
 ROEVS=ROE*X(J,1)*S
 QS=0.5*ROEVS*X(J,1)
 CALL AERO(FMACH,CLA,CD0,CLN,DCLDM,DCLNM,DCDM)
 DLDADV=ROEVS*(CLA+0.5*FMACH*DCLDM)
 DTDV=DTM/A
 DTDH=DTH
 DTDVDV=0.0
 DTDVDH=-DTDV/H1
 DTDHDH=-DTH/H1
 DLDA=QS*CLA
 DLDADA=0.0
 DDDADA=2.*QS*CLN
 DLDADH=-DLDA/H1
 DDDA=2.*QS*CLN*X(J,11)
 DDDADH=-DDDA/H1
 VDDADV=2.*QS*X(J,11)*(2.*CLN+FMACH*DCLNM)
 DDDADV=VDDADV/X(J,1)
 DLDV=DLDADV*X(J,11)
 DLDVDH=-DLDV/H1
```

```
 DLDVDV=(1.0/X(J,1))*(DLDV+1.5*ROEVS*FMACH*DCLDM*X(J,11))
 FL=QS*CLA*X(J,11)
 DLDH=-FL/H1
 DLDHDH=FL/(H1*H1)
 D=QS*(CD0+CLN*X(J,11)*X(J,11))
 DDDH=-D/H1
 DDDHDH=D/(H1*H1)
 DDDV=(2.*D/X(J,1))+.5*ROEVS*FMACH*(DCDM+DCLNM*X(J,11)*X(J,11))
 DDDVDH=-DDDV/H1
 DDDVDV=(1.0/X(J,1))*(DDDV+1.5*ROEVS*FMACH*(DCDM
1+DCLNM*X(J,11)*X(J,11)))
 F1=-GSG+(T*CA-D)/X(J,5)
 F2=(-GCG+(FL+T*SA)/X(J,5))/X(J,1)
 F3=X(J,1)*SG
 F4=X(J,1)*CG
 F5=-T/CGRAV
 F6=(-1./X(J,5))*(X(J,6)*(CA*DTDV-DDDV)+X(J,7)*(DLDV+DTDV*SA-F2
1*X(J,5))/X(J,1))-X(J,8)*SG-X(J,9)*CG+X(J,10)*DTDV/CGRAV
 F7=GCG*X(J,6)-(GSG*X(J,7)/X(J,1))-X(J,8)*X(J,1)*CG
1+X(J,9)*X(J,1)*SG
 F8=((-X(J,6)*(DTDH*CA-DDDH)-X(J,7)*(DLDH+DTDH*SA)/X(J,1))/
1X(J,5))+X(J,10)*DTDH/CGRAV
 F9=0.0
 F10=(-X(J,6)*(D-T*CA)+X(J,7)*(FL+T*SA)/X(J,1))/(X(J,5)*
1X(J,5))
 D1=-(T*SA+DDDA)/X(J,5)
 D2=(DLDA+T*CA)/(X(J,5)*X(J,1))
 D6=(X(J,6)*(DTDV*SA+DDDADV)-X(J,7)*(X(J,1)*(DLDADV+DTDV*CA)
1-(DLDA+T*CA))/(X(J,1)*X(J,1)))/X(J,5)
 D8=(X(J,6)*(DTDH*SA+DDDADH)-X(J,7)*(DLDADH+DTDH*CA)/X(J,1))/
1X(J,5)
 D10=(-X(J,6)*(DDDA+T*SA)+X(J,7)*(DLDA+T*CA)/X(J,1))/(X(J,5)*
1X(J,5))
 DF1DV=(DTDV*CA-DDDV)/X(J,5)
 DF2DV=((DLDV+DTDV*SA)/(X(J,5)*X(J,1)))-F2/X(J,1)
 DF3DV=SG
 DF4DV=CG
 DF5DV=-DTDV/CGRAV
 DF6DV=(-X(J,6)*(CA*DTDVDV-DDDVDV)/X(J,5))+(X(J,10)*DTDVDV/
1CGRAV)-(2.*X(J,7)/(X(J,5)*X(J,1)*X(J,1)))*(-(DLDV+DTDV*SA)+
1.5*X(J,1)*(DLDVDV+DTDVDV*SA)+X(J,5)*F2)
 DF7DV=(X(J,7)*GRAV*SG/(X(J,1)*X(J,1)))-X(J,8)*CG+X(J,9)*SG
 DF8DV=-(X(J,6)/X(J,5))*(DTDVDH*CA-DDDVDH)-(X(J,7)
1/(X(J,5)*X(J,1)*X(J,1)))*(X(J,1)*(DLDVDH+DTDVDH*SA)-
1(DLDH+DTDH*SA))+X(J,10)*DTDVDH/CGRAV
 DF10DV=(-X(J,6)*(DDDV-DTDV*CA)+X(J,7)*(X(J,1)*(DLDV+DTDV*SA)
1-(FL+T*SA))/(X(J,1)*X(J,1)))/(X(J,5)*X(J,5))
 DF1DG=-GRAV*CG
 DF2DG=GRAV*SG/X(J,1)
 DF3DG=X(J,1)*CG
 DF4DG=-X(J,1)*SG
 DF6DG=-X(J,8)*CG+X(J,9)*SG+X(J,7)*GRAV*SG/(X(J,1)*X(J,1))
 DF7DG=-X(J,6)*GRAV*SG-(X(J,7)*GRAV*CG/X(J,1))+X(J,8)*X(J,1)
1*SG+X(J,9)*X(J,1)*CG
```

```
      DF1DH=(DTDH*CA-DDDH)/X(J,5)
      DF2DH=(DLDH+DTDH*SA)/(X(J,5)*X(J,1))
      DF5DH=-DTDH/CGRAV
      DF6DH=-(X(J,6)/X(J,5))*(CA*DTDVDH-DDDVDH)-(X(J,7)
     1/(X(J,5)*X(J,1)*X(J,1)))*(X(J,1)*(DLDVDH+DTDVDH*SA)-
     1(DLDH+DTDH*SA))+X(J,10)*DTDVDH/CGRAV
      DF8DH=-(X(J,6)/X(J,5))*(DTDHDH*CA-DDDHDH)-
     1(X(J,7)/(X(J,5)*X(J,1)))*(DLDHDH+DTDHDH*SA)+X(J,10)*DTDHDH/CGRAV
      DF10DH=(-X(J,6)*(DDDH-DTDH*CA)+X(J,7)*(DLDH+DTDH*SA)/X(J,1))
     1/(X(J,5)*X(J,5))
      DF1DM=-(T*CA-D)/(X(J,5)*X(J,5))
      DF2DM=-((FL+T*SA)/X(J,1))/(X(J,5)*X(J,5))
      DF6DM=(1.0/(X(J,5)*X(J,5)))*(X(J,6)*(CA*DTDV-DDDV)+
     1(X(J,7)/(X(J,1)*X(J,1)))*(X(J,1)*(DLDV+DTDV*SA)-(FL+T*SA)))
      DF8DM=(X(J,6)*(DTDH*CA-DDDH)+(X(J,7)*(DLDH+DTDH*SA)/X(J,1)))/
     1(X(J,5)*X(J,5))
      DF10DM=2.*(X(J,6)*(D-T*CA)-X(J,7)*(FL+T*SA)/X(J,1))/(X(J,5)
     1*X(J,5)*X(J,5))
      DF6DLV=-(DTDV*CA-DDDV)/X(J,5)
      DF7DLV=GRAV*CG
      DF8DLV=-(DTDH*CA-DDDH)/X(J,5)
      DF10DLV=-(D-T*CA)/(X(J,5)*X(J,5))
      DF6DLG=-(((DLDV+DTDV*SA)/X(J,5))-F2)/X(J,1)
      DF7DLG=-GRAV*SG/X(J,1)
      DF8DLG=-(DLDH+DTDH*SA)/(X(J,5)*X(J,1))
      DF10DLG=(FL+T*SA)/(X(J,1)*X(J,5)*X(J,5))
      DF6DLH=-SG
      DF7DLH=-X(J,1)*CG
      DF6DLX=-CG
      DF7DLX=X(J,1)*SG
      DF6DLM=DTDV/CGRAV
      DF8DLM=DTDH/CGRAV
   11 CONTINUE
      DG1DA=1.0+(-X(J,6)*(T*CA+DDDADA)+X(J,7)*(DLDADA-T*SA)
     1/X(J,1))/X(J,5)
      Z=-DG1DA*1000.
      DG1DV=(-X(J,6)*(DTDV*SA+DDDADV)+(X(J,7)/(X(J,1)*X(J,1)))
     1*(X(J,1)*(DLDADV+CA*DTDV)-(DLDA+T*CA)))/X(J,5)
      DG1DH=(-X(J,6)*(SA*DTH+DDDADH)+(X(J,7)/X(J,1))*
     1(DLDADH+CA*DTH))/X(J,5)
      DG1DLV=-(T*SA+DDDA)/X(J,5)
      DG1DLG=(DLDA+T*CA)/(X(J,5)*X(J,1))
      G1=X(J,11)+(-X(J,6)*(T*SA+DDDA)+X(J,7)*(DLDA+T*CA)/X(J,1))/X(J,5)
      DG1DM=(X(J,11)-G1)/X(J,5)
      E1=-(X(J,1)*DG1DV+X(J,3)*DG1DH+X(J,5)*DG1DM
     1+X(J,6)*DG1DLV+X(J,7)*DG1DLG-G1)
      D1=D1/Z
      D2=D2/Z
      D6=D6/Z
      D8=D8/Z
      D10=D10/Z
      C(I,1,1)=DF1DV+D1*DG1DV
      C(I,1,2)=DF1DG
      C(I,1,3)=DF1DH+D1*DG1DH
```

```
C(I,1,5)=DF1DM+D1*DG1DM
C(I,1,6)=D1*DG1DLV
C(I,1,7)=D1*DG1DLG
C(I,2,1)=DF2DV+D2*DG1DV
C(I,2,2)=DF2DG
C(I,2,3)=DF2DH+D2*DG1DH
C(I,2,5)=DF2DM+D2*DG1DM
C(I,2,6)=D2*DG1DLV
C(I,2,7)=D2*DG1DLG
C(I,3,1)=DF3DV
C(I,3,2)=DF3DG
C(I,4,1)=DF4DV
C(I,4,2)=DF4DG
C(I,5,1)=DF5DV
C(I,5,3)=DF5DH
C(I,6,1)=DF6DV+D6*DG1DV
C(I,6,2)=DF6DG
C(I,6,3)=DF6DH+D6*DG1DH
C(I,6,5)=DF6DM+D6*DG1DM
C(I,6,6)=DF6DLV+D6*DG1DLV
C(I,6,7)=DF6DLG+D6*DG1DLG
C(I,6,8)=DF6DLH
C(I,6,9)=DF6DLX
C(I,6,10)=DF6DLM
C(I,7,1)=DF7DV
C(I,7,2)=DF7DG
C(I,7,6)=DF7DLV
C(I,7,7)=DF7DLG
C(I,7,8)=DF7DLH
C(I,7,9)=DF7DLX
C(I,8,1)=DF8DV+D8*DG1DV
C(I,8,3)=DF8DH+D8*DG1DH
C(I,8,5)=DF8DM+D8*DG1DM
C(I,8,6)=DF8DLV+D8*DG1DLV
C(I,8,7)=DF8DLG+D8*DG1DLG
C(I,8,10)=DF8DLM
C(I,10,1)=DF10DV+D10*DG1DV
C(I,10,3)=DF10DH+D10*DG1DH
C(I,10,5)=DF10DM+D10*DG1DM
C(I,10,6)=DF10DLV+D10*DG1DLV
C(I,10,7)=DF10DLG+D10*DG1DLG
E(I,1)=F1+D1*E1-(X(J,1)*DF1DV+X(J,2)*DF1DG+X(J,3)*DF1DH+X(J,5)
1*DF1DM)
E(I,2)=F2+D2*E1-(X(J,1)*DF2DV+X(J,2)*DF2DG+X(J,3)*DF2DH+X(J,5)
1*DF2DM)
E(I,3)=F3-(X(J,1)*DF3DV+X(J,2)*DF3DG)
E(I,4)=F4-(X(J,1)*DF4DV+X(J,2)*DF4DG)
E(I,5)=F5-(X(J,1)*DF5DV+X(J,3)*DF5DH)
E(I,6)=F6+D6*E1-(X(J,1)*DF6DV+X(J,2)*DF6DG+X(J,3)*DF6DH
1+X(J,5)*DF6DM+X(J,6)*DF6DLV+X(J,7)*DF6DLG+X(J,8)*DF6DLH
1+X(J,9)*DF6DLX+X(J,10)*DF6DLM)
E(I,7)=F7-(X(J,1)*DF7DV+X(J,2)*DF7DG+X(J,6)*DF7DLV
1+X(J,7)*DF7DLG+X(J,8)*DF7DLH+X(J,9)*DF7DLX)
E(I,8)=F8+D8*E1-(X(J,1)*DF8DV+X(J,3)*DF8DH+X(J,5)*
```

```
     1DF8DM+X(J,6)*DF8DLV+X(J,7)*DF8DLG+X(J,10)*DF8DLM)
      E(I,9)=0.0
      E(I,10)=F10+D10*E1-(X(J,1)*DF10DV+X(J,3)*DF10DH+X(J,5)*DF10DM+
     1X(J,6)*DF10DLV+X(J,7)*DF10DLG)
      GO TO (4,5,6),L
    4 CONTINUE
      CALL RPI(N,NV,NU,X,K,IT,IRNR)
C     WRITE(4,*)(X(3,I),I=1,NV)
      DO 8 I=1,NV
      X(2,I)=(X(1,I)+X(3,I))/2.
      GO TO (8,70)IP
   70 WRITE(4,*)'K=',K,'X=',(X(JJJ,I),JJJ=1,3)
    8 CONTINUE
      I=2
      J=2
      L=2
      GO TO 7
    5 CONTINUE
      I=3
      J=K
      L=3
      GO TO 7
    6 CONTINUE
      M=K1
      K1=K
      K=M
      ITAG=2
C     WRITE(6,*)(((C(II,JJ,KK),KK=1,N),JJ=1,N),II=1,3)
      RETURN
      END
      SUBROUTINE ALPHA(J,X,Y,IACON)
      DIMENSION X(3,12),Y(13,12)
      C=X(J,11)
      H1=27300.
      S=530.
      DV=Y(1,1)-X(J,1)
      DH=Y(1,3)-X(J,3)
      DLV=Y(1,6)-X(J,6)
      DLG=Y(1,7)-X(J,7)
      WRITE(6,*)'DV=',DV,'DH=',DH,'DLV=',DLV,'DLG=',DLG
C     PAUSE
      CALL ROE1(X(J,3),ROE)
      CALL MACH(X(J,1),X(J,3),FMACH,A)
      CALL THRUST(FMACH,X(J,3),T,DTM,DTH)
      ROEVS=ROE*X(J,1)*S
      QS=0.5*ROEVS*X(J,1)
      CALL AERO(FMACH,CLA,CD0,CLN,DCLDM,DCLNM,DCDM)
      DLDADV=ROEVS*(CLA+0.5*FMACH*DCLDM)
      DTDV=DTM/A
      DTDVDV=0.0
      DTDVDH=-DTDV/H1
      DTDHDH=-DTH/H1
      DLDA=QS*CLA
      DLDADA=0.0
```

```
      DDDADA=2.*QS*CLN
      DO 1 L=1,2
      M=1
      IF(L-1)4,4,5
    5 M=800
    4 CONTINUE
      DO 1 I=1,M
      IF(L-1)6,6,7
    7 CONTINUE
      IF(ABS(Y(1,11)-C)-.00001)2,2,3
    3 CONTINUE
      C=Y(1,11)
    6 CONTINUE
      WRITE(6,*)'C=',C
      DLDADH=-DLDA/H1
      DDDA=2.*QS*CLN*C
      DDDADH=-DDDA/H1
      VDDADV=2.*QS*C*(2.*CLN+FMACH*DCLNM)
      DDDADV=VDDADV/X(J,1)
      SA=SIN(C)
      CA=COS(C)
      DG1DA=-X(J,6)*X(J,1)*(T*CA+DDDADA)+X(J,7)*(DLDADA-T*SA)
      Z=-DG1DA*10.
      DG1DV=-X(J,6)*(T*SA+DDDA+FMACH*SA*DTM+VDDADV)+
     1X(J,7)*(DLDADV+CA*DTDV)
      DG1DH=-X(J,6)*X(J,1)*(SA*DTH+DDDADH)+X(J,7)
     1*(DLDADH+CA*DTH)
      DG1DLV=-X(J,1)*(T*SA+DDDA)
      DG1DLG=DLDA+T*CA
      G1=-X(J,6)*X(J,1)*(T*SA+DDDA)+X(J,7)*(DLDA+T*CA)
      Y(1,11)=C+.2*(DG1DV*DV+DG1DH*DH+DG1DLV*Y(1,6)
     1+DG1DLG*Y(1,7))/Z
      GO TO 80
      SIGNY=1.0
      SIGNC=1.0
      IF(Y(1,11))70,80,71
70    SIGNY=-1.0
71    IF(SIGNC)72,80,73
72    SIGNC=-1.0
73     SIGN=SIGNY*SIGNC
      IF(SIGN)74,80,80
74    Y(1,11)=(Y(1,11)+.8*C)/2.0
80     WRITE(6,*)'Y11=',Y(1,11),'Z=',Z,'G1DV=',DG1DV,'G1DH=',DG1DH
    1 CONTINUE
      IACON=1
      GO TO 40
    2 CONTINUE
      IACON=0
   40 WRITE(6,*)C
      DLDV=DLDADV*C
      DLDVDH=-DLDV/H1
      DLDVDV=(1.0/X(J,1))*(DLDV+ROEVS*FMACH*DCLDM*C)
      FL=QS*CLA*C
      DLDH=-FL/H1
```

```
      DLDHDH=FL/(H1*H1)
      D=QS*(CD0+CLN*C*C)
      DDDH=-D/H1
      DDDHDH=D/(H1*H1)
      DDDV=(2.*D/X(J,1))+.5*ROEVS*FMACH*(DCDM+DCLNM*C*C)
      DDDVDH=-DDDV/H1
      DDDVDV=(1.0/X(J,1))*(DDDV+ROEVS*FMACH*(DCDM+DCLNM*C*C))
      WRITE(6,*)'FL=',FL,'DRAG=',D,'ALPHA=',C
C     PAUSE
      RETURN
      END
      SUBROUTINE ALPHA1(J,X,Y,IACON)
      DIMENSION X(3,12),Y(13,12)
      C=X(J,11)
C     Y(1,11)=C
C     RETURN
      H1=27300.
      S=530.
      CALL ROE1(Y(1,3),ROE)
      CALL MACH(Y(1,1),Y(1,3),FMACH,A)
      CALL THRUST(FMACH,Y(1,3),T,DTM,DTH)
      ROEVS=ROE*Y(1,1)*S
      QS=0.5*ROEVS*Y(1,1)
      CALL AERO(FMACH,CLA,CD0,CLN,DCLDM,DCLNM,DCDM)
      DLDA=QS*CLA
      DLDADA=0.0
      DDDADA=2.*QS*CLN
      DO 1 L=1,2
      M=1
      IF(L-1)4,4,5
    5 M=800
    4 CONTINUE
      DO 1 I=1,M
      IF(L-1)6,6,7
    7 CONTINUE
      IF(ABS(Y(1,11)-C)-.00001)2,2,3
    3 CONTINUE
      C=Y(1,11)
    6 CONTINUE
      WRITE(6,*)'C=',C
      DDDA=2.*QS*CLN*C
      SA=SIN(C)
      CA=COS(C)
      DG1DA=-Y(1,6)*Y(1,1)*(T*CA+DDDADA)+Y(1,7)*(DLDADA-T*SA)
      Z=-DG1DA*10.
      G1=-Y(1,6)*Y(1,1)*(T*SA+DDDA)+Y(1,7)*(DLDA+T*CA)
      Y(1,11)=C+.8*(G1)/Z
80    WRITE(6,*)'Y11=',Y(1,11),'Z=',Z,'G1=',G1,'T=',T
      WRITE(6,*)'V=',Y(1,1),'LV=',Y(1,6),'LG=',Y(1,7)
      WRITE(6,*)'D2DA=',DDDADA,'DDA=',DDDA,'DLDA=',DLDA
      WRITE(6,*)'SA=',SA,'CA=',CA
    1 CONTINUE
      IACON=1
      GO TO 40
```

```
  2 CONTINUE
    GO TO 41
    SIGNC=1.
    IF(C)50,41,51
 50 SIGNC=-1.
 51 IF(ABS(C)-.4)41,41,53
 53 C=SIGNC*.1
 41 IACON=0
 40 WRITE(6,*)C
    Y(1,11)=C
    FL=QS*CLA*C
    D=QS*(CD0+CLN*C*C)
    WRITE(6,*)'FL=',FL,'DRAG=',D,'ALPHA=',C
C     PAUSE
    RETURN
    END
    SUBROUTINE ALPHA2(J,X,Y,IACON)
    DIMENSION X(3,12),Y(13,12)
    C=X(J,11)
    H1=27300.
    S=530.
    DM=Y(1,5)-X(J,5)
    DV=Y(1,1)-X(J,1)
    DH=Y(1,3)-X(J,3)
    DLV=Y(1,6)-X(J,6)
    DLG=Y(1,7)-X(J,7)
    WRITE(6,*)'DV=',DV,'DH=',DH,'DLV=',DLV,'DLG=',DLG
C     PAUSE
    CALL ROE1(X(J,3),ROE)
    CALL MACH(X(J,1),X(J,3),FMACH,A)
    CALL THRUST(FMACH,X(J,3),T,DTM,DTH)
    ROEVS=ROE*X(J,1)*S
    QS=0.5*ROEVS*X(J,1)
    CALL AERO(FMACH,CLA,CD0,CLN,DCLDM,DCLNM,DCDM)
    DLDADV=ROEVS*(CLA+0.5*FMACH*DCLDM)
    DTDV=DTM/A
    DTDVDV=0.0
    DTDVDH=-DTDV/H1
    DTDHDH=-DTH/H1
    DLDA=QS*CLA
    DLDADA=0.0
    DDDADA=2.*QS*CLN
    DO 1 L=1,1
C   DO 1 L=1,2
    M=1
    IF(L-1)4,4,5
  5 M=800
  4 CONTINUE
    DO 1 I=1,M
    IF(L-1)6,6,7
  7 CONTINUE
    IF(ABS(Y(1,11)-C)-.00001)2,2,3
  3 CONTINUE
    C=Y(1,11)
```

```
  6 CONTINUE
    WRITE(6,*)'C=',C
    DLDADH=-DLDA/H1
    DDDA=2.*QS*CLN*C
    DDDADH=-DDDA/H1
    VDDADV=2.*QS*C*(2.*CLN+FMACH*DCLNM)
    DDDADV=VDDADV/X(J,1)
    SA=SIN(C)
    CA=COS(C)
    DG1DA=1.0+(-X(J,6)*(T*CA+DDDADA)+X(J,7)*(DLDADA-T*SA)
   1/X(J,1))/X(J,5)
    Z=-DG1DA*1000.
    DG1DV=(-X(J,6)*(DTDV*SA+DDDADV)+(X(J,7)/(X(J,1)*X(J,1)))
   1*(X(J,1)*(DLDADV+CA*DTDV)-(DLDA+T*CA)))/X(J,5)
    DG1DH=(-X(J,6)*(SA*DTH+DDDADH)+(X(J,7)/X(J,1))*
   1(DLDADH+CA*DTH))/X(J,5)
    DG1DLV=-(T*SA+DDDA)/X(J,5)
    DG1DLG=(DLDA+T*CA)/(X(J,5)*X(J,1))
    G1=X(J,11)+(-X(J,6)*(T*SA+DDDA)+X(J,7)*(DLDA+T*CA)/X(J,1))/X(J,5)
    DG1DM=(X(J,11)-G1)/X(J,5)
    E1=-(X(J,1)*DG1DV+X(J,3)*DG1DH-G1)
    Y(1,11)=C+(DG1DV*DV+DG1DH*DH+DG1DLV*DLV
   1+DG1DLG*DLG+DG1DM*DM+G1)/Z
    GO TO 80
 80 WRITE(6,*)'Y11=',Y(1,11),'Z=',Z,'G1DV=',DG1DV,'G1DH=',DG1DH
  1 CONTINUE
    IACON=1
    GO TO 40
  2 CONTINUE
    IACON=0
 40 WRITE(6,*)C
    DLDV=DLDADV*C
    DLDVDH=-DLDV/H1
    DLDVDV=(1.0/X(J,1))*(DLDV+ROEVS*FMACH*DCLDM*C)
    FL=QS*CLA*C
    DLDH=-FL/H1
    DLDHDH=FL/(H1*H1)
    D=QS*(CD0+CLN*C*C)
    DDDH=-D/H1
    DDDHDH=D/(H1*H1)
    DDDV=(2.*D/X(J,1))+.5*ROEVS*FMACH*(DCDM+DCLNM*C*C)
    DDDVDH=-DDDV/H1
    DDDVDV=(1.0/X(J,1))*(DDDV+ROEVS*FMACH*(DCDM+DCLNM*C*C))
    WRITE(6,*)'FL=',FL,'DRAG=',D,'ALPHA=',C
C     PAUSE
    RETURN
    END
    SUBROUTINE AERO(FMACH,CLA,CD0,CLN,DCLAM,DCLNM,DCDM)
    DIMENSION RM(10)
    OPEN(11,FILE='LOG.CLD',STATUS='UNKNOWN',ACCESS='SEQUENTIAL',
   1 FORM='FORMATTED')
  4 FORMAT(F10.4,4F10.4)
    FMACHE=FMACH
    IF(FMACHE)2,3,3
```

```
  2 FMACH=0.0
  3 RM(1)=.0
    RM(2)=.4
    RM(3)=.8
    RM(4)=.9
    RM(5)=1.0
    DO 10 I=6,10
 10 RM(I)=RM(I-1)+.2
    DO 11 I=1,9
    IF(FMACH-RM(I))12,11,11
 11 CONTINUE
    I=I-1
 12 DM=FMACH-RM(I-1)
    DM1=RM(I)-RM(I-1)
    FM1=RM(I-1)
    FM2=RM(I)
    REWIND 11
 14 READ (11,4,END=20)FM,CL1,CD1,FN1,CLN1
    TEST=ABS(FM1-FM)
    IF (TEST-.01)13,13,14
 13 READ (11,4,END=20)FM,CL2,CD2,FN2,CLN2
    DCLAM=(CL2-CL1)/DM1
    DCDM=(CD2-CD1)/DM1
    DCLNM=(CLN2-CLN1)/DM1
    CLA=CL1+DCLAM*DM
    CD0=CD1+DCDM*DM
    CLN=CLN1+DCLNM*DM
C     WRITE(6,*)'CLA=',CLA,'CD0=',CD0,'CLN=',CLN
C     WRITE(6,*)'DCLAM=',DCLAM,'DCDM=',DCDM,'DCLNM=',DCLNM
    FMACH=FMACHE
    RETURN
 20 WRITE(6,*)'READ ERROR UNIT 11'
    STOP
    END
        SUBROUTINE SOLVX(A,B,X,N,V)
        DIMENSION A(12,12),B(12),X(12)
        NM1=N-1
        DO 2 J=1,N
        JP1=J+1
        JM1=J-1
        DO 6 I=J,N
        ASUM=0.
        IF(JM1)6,6,7
7       DO 9 K=1,JM1
9       ASUM=ASUM+A(I,K)*A(K,J)
6       A(I,J)=A(I,J)-ASUM
        AMAX=A(J,J)
        IMAX=J
        IF(JP1-N)20,20,21
20      CONTINUE
        DO 1 I=JP1,N
        IF(ABS(AMAX)-ABS(A(I,J)))3,1,1
3       AMAX=A(I,J)
        IMAX=I
```

```
1       CONTINUE
21      CONTINUE
        IF(ABS(AMAX)-1.E-30)10,10,4
10      V=1.
        RETURN
4       DO 5 K=1,N
        ASAVE=A(IMAX,K)
        A(IMAX,K)=A(J,K)
5       A(J,K)=ASAVE
        ASAVE=B(IMAX)
        B(IMAX)=B(J)
        B(J)=ASAVE
        I1=J
        J1=JP1
        IF(JP1-N)22,22,23
22      CONTINUE
        DO 8 J2=JP1,N
        ASUM=0.
        IF(JM1)8,8,11
11      DO 12 K=1,JM1
12      ASUM=ASUM+A(I1,K)*A(K,J2)
8       A(I1,J2)=(A(I1,J2)-ASUM)/A(I1,I1)
23      CONTINUE
        ASUM=0.
        IF(JM1)2,2,13
13      DO 14 K=1,JM1
14      ASUM=ASUM+A(I1,K)*B(K)
2       B(I1)=(B(I1)-ASUM)/A(I1,I1)
        DO 15 J=1,N
        I1=N-J+1
        I=I1+1
        ASUM=0.
        IF(I1-N)17,15,15
17      DO 16 K=I,N
16      ASUM=ASUM+A(I1,K)*X(K)
15      X(I1)=B(I1)-ASUM
        V=0.
        RETURN
        END
        SUBROUTINE INTRNG(A,C,E,N,NH,DT)
        DIMENSION A(13,12),C(3,12,12),E(3,12),D(4,12)
        NIT=NH+1
        DO 1 K=1,NIT
        DO 100 I=1,N
        TEMP=0.
        IF(K-1)3,2,3
2       TEMP=E(1,I)
3       CONTINUE
        DO 10 J=1,N
10      TEMP=TEMP+C(1,I,J)*A(K,J)
100     D(1,I)=DT*TEMP
        DO 110 KK=1,2
        DO 110 I=1,N
        TEMP=0.
```

```
      IF(K-1)5,4,5
4     TEMP=E(2,I)
5     CONTINUE
      DO 11 J=1,N
11    TEMP=TEMP+C(2,I,J)*(A(K,J)+D(KK,J)/2.)
110   D(KK+1,I)=DT*TEMP
      DO 130 I=1,N
      TEMP=0.
      IF(K-1)7,6,7
6     TEMP=E(3,I)
7     CONTINUE
      DO 13 J=1,N
13    TEMP=TEMP+C(3,I,J)*(A(K,J)+D(3,J))
130   D(4,I)=DT*TEMP
      DO 200 I=1,N
200   A(K,I)=A(K,I)+(D(1,I)+2.*D(2,I)+2.*D(3,I)+D(4,I))/6.
1     CONTINUE
      RETURN
      END
   SUBROUTINE WYTIN(A,NU,IR,IRN,IT,N,NV,NH)
   DIMENSION A(13,12)
   J=NH+1
   WRITE(NU,REC=IR)N,J,IT,((A(K,I),I=1,N),K=1,J)
   IRN=IR+1
   RETURN
   END
   SUBROUTINE RYTIN(A,NU,IR,IRN,IT,N,NV,NH)
   DIMENSION A(13,12)
   J=NH+1
   READ(NU,REC=IR)N,J,IT,((A(K,I),I=1,N),K=1,J)
   IRN=IR+1
   RETURN
   END
   SUBROUTINE MODIC(N,NV,NH,IR,IRITS,ICUS,NU,TCD,A)
   DIMENSION A(13,12),P(12),B(12),C(12),H(12,12)
   DIMENSION TCD(12)
   DIMENSION ICUS(12)
 25 FORMAT(12E14.4)
   CALL RYTIN(A,NU,IR,IRN,IT,N,NV,NH)
   DO 2 I=1,NH
   DO 3 K=1,NH
   L=ICUS(I)
   H(I,K)=A((K+1),L)
  3 CONTINUE
   P(I)=A(1,L)
   B(I)=TCD(I)-P(I)
  2 CONTINUE
   WRITE(4,25)(TCD(I),P(I),(H(I,K),K=1,NH),B(I),I=1,NH)
   CALL SOLVX(H,B,C,NH,V)
   WRITE(4,25)(C(I),I=1,NH)
   NF=1
   CALL RYTIN(A,NU,IRITS,IRN,IT,N,NV,NH)
   NN=NH+1
   DO 5 K=1,N
```

```
  5 WRITE(4,25)(A(I,K),I=1,NN)
   DO 4 I=1,NH
   L=I+5
   DO 4 K=1,NH
   A(1,L)=A(1,L)+C(K)*A(K+1,L)
  4 CONTINUE
   RETURN
   END
       SUBROUTINE CONVT(N,NV,NU,NT,ICON)
       DIMENSION A(3,12),EMAX(12)
10     FORMAT(' ERR')
11     FORMAT(1E14.4)
       TEST=.005
       ICON=0.
       SUM=0.
       IRNR1=1
       IRNR2=2
       J=1
       K=2
       DO 9 L=1,N
       EMAX(L)=0.
9      CONTINUE
       DO 1 I=1,NT+1
       CALL RPI(N,NV,NU,A,J,IT1,IRNR1)
       CALL RPI(N,NV,NU,A,K,IT2,IRNR2)
       IF(IT1-IT2)3,2,3
3      WRITE(4,10)
2      CONTINUE
       DO 4 L=1,N
       E=ABS(A(1,L)-A(2,L))
       IF(E-EMAX(L))4,4,7
7      EMAX(L)=E
4      CONTINUE
1      CONTINUE
       DO 8 L=1,N
       SUM=SUM+EMAX(L)
8      CONTINUE
       WRITE(4,11)SUM
       IF(ABS(SUM)-TEST)5,5,6
5      ICON=1
6      RETURN
       END
       SUBROUTINE WYOT(N,NV,NU,NT,M)
       DIMENSION A(3,12)
10     FORMAT(1I4,9E14.4)
       L=1
       IX=1
       IY=NV
       IF(N-9)1,1,2
2      IY=9
       L=2
1      K=1
       DO 3 J=1,L
       GO TO (4,5),J
```

```
5       IX=10
        IY=NV
4       IRN=M
        DO 6 I=1,NT+1
        CALL RPI(N,NV,NU,A,K,IT1,IRN)
        WRITE (4,10)IT1,(A(1,IJ),IJ=IX,IY)
6       CONTINUE
3       CONTINUE
        RETURN
        END
        SUBROUTINE WPI(N,NV,NU,A,K,IT,IRNW)
        DIMENSION A(3,12)
        IRC=IRNW
        WRITE(NU,REC=IRC)IT,(A(K,I),I=1,NV)
        IRNW=IRNW+2
        RETURN
        END
        SUBROUTINE RPI(N,NV,NU,A,K,IT,IRNR)
        DIMENSION A(3,12)
        IRC=IRNR
        READ(NU,REC=IRC)IT,(A(K,I),I=1,NV)
        IRNR=IRNR+2
        RETURN
        END
        SUBROUTINE DAUX2(N,NV,NU,NUH,Y,L,A,IRITS,IRN,NH,ICUS,ITAG3)
        DIMENSION Y(13,12),A(3,12)
     DIMENSION ICUS(12)
        IRNR=L
        K=1
        CALL RPI(N,NV,NU,A,K,IT,IRNR)
        NIT=NH+1
        DO 1 I=2,NIT
     M=I+4
        DO 1 J=1,N
        Y(I,J)=0.
        IF(J-M)3,2,3
2       Y(I,J)=1.
3       CONTINUE
1       CONTINUE
        DO 4 I=1,N
        Y(1,I)=A(K,I)
4       CONTINUE
     IF(ITAG3)10,10,11
10     Y(1,6)=-17.73425
     Y(1,7)=-1530.7384
     Y(1,8)=-.052362
     Y(1,9)=.0
     Y(1,10)=52.623
11      IT=0
        WRITE(4,*)IT,(Y(1,I),I=1,N)
     CALL WYTIN(Y,NUH,IRITS,IRN,IT,N,NV,NH)
        RETURN
        END
        SUBROUTINE ZCE(N,C,E)
```

```
      DIMENSION C(3,12,12),E(3,12)
      DO 1 I=1,3
      DO 1 J=1,N
      E(I,J)=0.
      DO 1 K=1,N
      C(I,J,K)=0.
1     CONTINUE
      RETURN
      END
      SUBROUTINE GSO(N,NV,NH,NUH,IRL,IRITS,Y)
      DIMENSION A(12,12),B(12,12),Y(13,12),F(13,12),S(12,12)
      CALL RYTIN(F,NUH,IRITS,IRN,IT,N,NV,NH)
      DO 1 I=1,NH
      DO 1 J=I,NH
      S(J,I)=0.0
1     CONTINUE
      DO 8 I=1,NH
      DO 8 J=1,N
      A(I,J)=Y(I+1,J)
      B(I,J)=F(I+1,J)
8     CONTINUE
C     DO 20 J=1,N
C20   WRITE(4,*)'Y=',(Y(I,J),I=1,NH+1)
C     DO 21 J=1,N
C21   WRITE(4,*)'F=',(F(I,J),I=1,NH+1)
      DO 2 I=1,NH
      DO 3 L=1,I-1
C     WRITE(6,*)L
C     PAUSE
      DO 3 K=1,N
C     WRITE(6,*)'K=',K
      A(I,K)=A(I,K)-S(I,L)*A(L,K)
      B(I,K)=B(I,K)-S(I,L)*B(L,K)
3     CONTINUE
      DO 4 J=I,NH
      DO 4 K=1,N
      S(J,I)=S(J,I)+A(J,K)*A(I,K)
4     CONTINUE
      IF(I-NH)9,10,9
9     CONTINUE
      DO 5 J=I+1,NH
      S(J,I)=S(J,I)/S(I,I)
5     CONTINUE
10    CONTINUE
2     CONTINUE
C     WRITE(6,*)(S(I,I),I=1,NH)
      DO 6 I=1,NH
      DO 6 K=1,N
      A(I,K)=A(I,K)/SQRT(S(I,I))
      B(I,K)=B(I,K)/SQRT(S(I,I))
6     CONTINUE
      DO 7 I=1,NH
      DO 7 K=1,N
      Y(I+1,K)=A(I,K)
```

```
      F(I+1,K)=B(I,K)
7     CONTINUE
      CALL WYTIN(F,NUH,IRITS,IRN,IT,N,NV,NH)
      CALL WYTIN(Y,NUH,IRL,IRN,IT,N,NV,NH)
      RETURN
      END
```

Appendix D.

On-Line Identification And Adaptation Programme.

```
      DIMENSION TCD(15),X(3,15),A(3,15),Y(16,15),XX(12,2)
      DIMENSION E(3,15),C(3,15,15),XYZ(15),AA(15,15),BB(15)
      DIMENSION ICUS(15),P(4),PI(4),FK(4),G(4),GI(4)
      OPEN (4,FILE='OUT.DAT',STATUS='UNKNOWN',ACCESS='SEQUENTIAL',
     1 FORM='FORMATTED')
      OPEN (7,FILE='QD.DAT',STATUS='UNKNOWN',ACCESS='SEQUENTIAL',
     1 FORM='FORMATTED')
      OPEN(10,FILE='WNZTA.RES',STATUS='UNKNOWN',ACCESS='SEQUENTIAL',
     1 FORM='FORMATTED')
 100  FORMAT(I10,6E15.4)
 200  FORMAT(I10,E15.4)
      NU=7
      NUH=12
      IR=1
      N=14
      NN=4
      NV=5
      DT=.001
      NT=332
      CALL ZCE(N,C,E)
      NH1=0
      ITT=0
      DO 3 J=1,14
    3 Y(1,J)=0.
      I=0
      IT=0
      ICD4=0
      READ(7,200)ITI,QD
      QD=10.*QD
      CALL PARAM(IT,FK)
      FKC=-.035*4.
      FKI=20.
      TC=3.
      G(1)=.5
      G(2)=5.
      G(3)=-100.
      G(4)=-2.
 12   DO 7 K=1,NT
      DO 9 J=1,5
      DO 20 LL=1,2
      READ(7,200)ITI,QD
      QD=10.*QD
      DO 2 I=1,100
      ITT=ITT+1
      U=Y(1,3)+FKC*(QD-Y(1,1))
      CALL DAUX4(N,C,E,FK,G,QD,FKC,FKI,TC,U)
```

```
 19  CALL INTRNG(Y,C,E,N,NH1,DT)
     ICD4=ICD4+1
     IF(ICD4-3320)2,18,2
 18   IT=IT+1
     ICD4=0
     CALL PARAM(IT,FK)
  2      CONTINUE
     XX(1,LL)=Y(1,1)
     XX(2,LL)=Y(1,2)
     DO 21 I=5,13,2
     XX((I-2),LL)=Y(1,I)
     XX((I-1),LL)=Y(1,(I+1))
 21  CONTINUE
 20  CONTINUE
C RESET I/C FOR PI AND HOMOG.
     DO 22 I=7,14
     Y(1,I)=0.
 22  CONTINUE
C SOLVE FOR C'S
     AA(1,1)=XX(5,1)
     AA(1,2)=XX(7,1)
     AA(1,3)=XX(9,1)
     AA(1,4)=XX(11,1)
     AA(2,1)=XX(6,1)
     AA(2,2)=XX(8,1)
     AA(2,3)=XX(10,1)
     AA(2,4)=XX(12,1)
     AA(3,1)=XX(5,2)
     AA(3,2)=XX(7,2)
     AA(3,3)=XX(9,2)
     AA(3,4)=XX(11,2)
     AA(4,1)=XX(6,2)
     AA(4,2)=XX(8,2)
     AA(4,3)=XX(10,2)
     AA(4,4)=XX(12,2)
     BB(1)=XX(1,1)-XX(3,1)
     BB(2)=XX(2,1)-XX(4,1)
     BB(3)=XX(1,2)-XX(3,2)
     BB(4)=XX(2,2)-XX(4,2)
     DO 26 I=1,4
 26  WRITE(6,*)(AA(I,J),J=1,4),BB(I),XYZ(I),Y(1,1),Y(1,5)
     CALL SOLVX(AA,BB,XYZ,NN,V)
     DO 27 I=1,4
 27  WRITE(6,*)(XYZ(J),J=1,4)
     DO 24 I=1,4
     GI(I)=XYZ(I)+G(I)
 24  CONTINUE
     CALL WNZ(FK,P)
     CALL WNZ1(GI,PI,FKC,FKI,TC)
     WRITE(6,*)(P(I),PI(I),I=1,4)
     DO 25 I=1,4
     G(I)=GI(I)
 25  CONTINUE
     Y(1,5)=XX(1,2)
```

```
  Y(1,6)=XX(2,2)
9     CONTINUE
7     CONTINUE
  STOP
      END
 SUBROUTINE DAUX4(N,C,E,FK,G,QD,FKC,FKI,TC,U)
 DIMENSION C(3,15,15),E(3,15),FK(4),G(4)
 DO 1 J=1,3
 C(J,1,1)=-FK(1)
 C(J,1,2)=1.
 C(J,2,1)=-FK(2)
 C(J,3,3)=-1./TC
 C(J,3,4)=1./TC
 C(J,4,1)=-FKC*FKI
 C(J,5,5)=-G(1)
 C(J,5,6)=1.
 C(J,6,5)=-G(2)
 C(J,7,1)=-1.
 C(J,7,7)=-G(1)
 C(J,7,8)=1.
 C(J,8,7)=-G(2)
 C(J,9,9)=-G(1)
 C(J,9,10)=1.
 C(J,10,1)=-1.
 C(J,10,9)=-G(2)
 C(J,11,11)=-G(1)
 C(J,11,12)=1.
 C(J,12,11)=-G(2)
 C(J,13,13)=-G(1)
 C(J,13,14)=1.
 C(J,14,13)=-G(2)
 E(J,1)=FK(3)*U
 E(J,2)=FK(2)*FK(4)*U
 E(J,4)=FKC*FKI*QD
 E(J,5)=G(3)*U
 E(J,6)=G(4)*U
 E(J,11)=U
 E(J,14)=U
1 CONTINUE
 RETURN
 END
      SUBROUTINE INTRNG(A,C,E,N,NH,DT)
      DIMENSION A(16,15),C(3,15,15),E(3,15),D(4,15)
      NIT=NH+1
      DO 1 K=1,NIT
      DO 100 I=1,N
      TEMP=0.
      IF(K-1)3,2,3
2     TEMP=E(1,I)
3     CONTINUE
      DO 10 J=1,N
10    TEMP=TEMP+C(1,I,J)*A(K,J)
100   D(1,I)=DT*TEMP
```

```
      DO 110 KK=1,2
      DO 110 I=1,N
      TEMP=0.
      IF(K-1)5,4,5
4     TEMP=E(2,I)
5     CONTINUE
      DO 11 J=1,N
11    TEMP=TEMP+C(2,I,J)*(A(K,J)+D(KK,J)/2.)
110   D(KK+1,I)=DT*TEMP
      DO 130 I=1,N
      TEMP=0.
      IF(K-1)7,6,7
6     TEMP=E(3,I)
7     CONTINUE
      DO 13 J=1,N
13    TEMP=TEMP+C(3,I,J)*(A(K,J)+D(3,J))
130   D(4,I)=DT*TEMP
      DO 200 I=1,N
200   A(K,I)=A(K,I)+(D(1,I)+2.*D(2,I)+2.*D(3,I)+D(4,I))/6.
1     CONTINUE
      RETURN
      END
      SUBROUTINE ZCE(N,C,E)
      DIMENSION C(3,15,15),E(3,15)
      DO 1 I=1,3
      DO 1 J=1,N
      E(I,J)=0.
      DO 1 K=1,N
      C(I,J,K)=0.
1     CONTINUE
      RETURN
      END

      SUBROUTINE SOLVX(A,B,X,N,V)
      DIMENSION A(15,15),B(15),X(15)
      NM1=N-1
      DO 2 J=1,N
      JP1=J+1
      JM1=J-1
      DO 6 I=J,N
      ASUM=0.
      IF(JM1)6,6,7
7     DO 9 K=1,JM1
9     ASUM=ASUM+A(I,K)*A(K,J)
6     A(I,J)=A(I,J)-ASUM
      AMAX=A(J,J)
      IMAX=J
      IF(JP1-N)20,20,21
20    CONTINUE
      DO 1 I=JP1,N
      IF(ABS(AMAX)-ABS(A(I,J)))3,1,1
3     AMAX=A(I,J)
      IMAX=I
```

```
1       CONTINUE
21      CONTINUE
        IF(ABS(AMAX)-1.E-30)10,10,4
10      V=1.
        RETURN
4       DO 5 K=1,N
        ASAVE=A(IMAX,K)
        A(IMAX,K)=A(J,K)
5       A(J,K)=ASAVE
        ASAVE=B(IMAX)
        B(IMAX)=B(J)
        B(J)=ASAVE
        I1=J
        J1=JP1
        IF(JP1-N)22,22,23
22      CONTINUE
        DO 8 J2=JP1,N
        ASUM=0.
        IF(JM1)8,8,11
11      DO 12 K=1,JM1
12      ASUM=ASUM+A(I1,K)*A(K,J2)
8       A(I1,J2)=(A(I1,J2)-ASUM)/A(I1,I1)
23      CONTINUE
        ASUM=0.
        IF(JM1)2,2,13
13      DO 14 K=1,JM1
14      ASUM=ASUM+A(I1,K)*B(K)
2       B(I1)=(B(I1)-ASUM)/A(I1,I1)
        DO 15 J=1,N
        I1=N-J+1
        I=I1+1
        ASUM=0.
        IF(I1-N)17,15,15
17      DO 16 K=I,N
16      ASUM=ASUM+A(I1,K)*X(K)
15      X(I1)=B(I1)-ASUM
        V=0.
        RETURN
        END
```

```
      SUBROUTINE WNZ(A,R)
      DIMENSION A(4),R(4)
      WN=SQRT(A(2))
      ZETA=A(1)/(2.*WN)
      FKO=A(4)
      T2=A(3)/(A(2)*A(4))
      R(1)=WN
      R(2)=ZETA
      R(3)=FKO
      R(4)=T2
      RETURN
      END
```

```
 SUBROUTINE WNZ1(A,R,FKC,FKI,TC)
  DIMENSION A(4),R(4)
  WN=SQRT(A(2))
  ZETA=A(1)/(2.*WN)
  FKO=A(4)/A(2)
  T2=A(3)/A(4)
  R(1)=WN
  R(2)=ZETA
  R(3)=FKO
  R(4)=T2
  FKC=4.*2.*ZETA/(WN*FKO)
  FKI=WN/2.*ZETA
  TC=1./(2.*ZETA*WN)
  RETURN
  END
 SUBROUTINE PARAM(IT,FK)
 DIMENSION FK(4),RES(20)
 REWIND 10
5 READ(10,*)J,(RES(I),I=1,18)
 IF(J-IT)5,6,6
6 T2=RES(14)
 FK(1)=RES(11)
 FK(2)=RES(12)
 FK(4)=RES(17)
 FK(3)=FK(4)*T2*FK(2)
 RETURN
 END
```

Acknowledgements

The author acknowledges, with much gratitude, the financial support provided by both G.E.C. Avionics and The Defence Research Agency - formerly Elliott Brothers London Ltd, and The Royal Aircraft Establishment Farnborough respectively. Thanks are due specifically to W.H.Alexander and G. Belcher of G.E.C. and F.Gill of Farnborough.

The support of Professor D. Bell in iniating this research is also appreciated.

Acknowledgement is also due to the University of Bath for the opportunity to submit this thesis for the degree of Doctor of Philosophy, as a member of staff.

The author is extremely appreciative of the support and encouragement of colleagues, and is particularly grateful to both Professor J.F. Eastham and Dr. R.T. Lipczynski of the School of Electronic and Electrical Engineering for undertaking the internal assessment of this research work.

Finally, I recognise the support of my family, and in particular that of my wife Audrey, who has had to endure my occasional fractious moments while undertaking this research.